2

Carderock Division
Naval Surface Warfare Center
Bethesda, MD 20084-5000

CRDKNSWC/HD-0936-06 Feb 1994

Ship Hydromechanics Department

Departmental Report

USER'S MANUAL FOR THE SHIP MOTIONS PREDICTION APPLICATIONS MANAGER - PREDICT

by

T. C. Smith

C. Bennett

Approved for public release, distribution unlimited.

CRDKNSWC/HD-0936-06 User's Manual for the Ship Motions Prediction Applications Manager - PREDICT

MAJOR DTRC TECHNICAL COMPONENTS

CODE 011	DIRECTOR OF TECHNOLOGY, PLANS AND ASSESSMENT
12	SHIP SYSTEMS INTEGRATION DEPARTMENT
14	SHIP ELECTROMAGNETIC SIGNATURES DEPARTMENT
15	SHIP HYDROMECHANICS DEPARTMENT
16	AVIATION DEPARTMENT
17	SHIP STRUCTURES AND PROTECTION DEPARTMENT
18	COMPUTATION, MATHEMATICS & LOGISTICS DEPARTMENT
19	SHIP ACOUSTICS DEPARTMENT
27	PROPULSION AND AUXILIARY SYSTEMS DEPARTMENT
28	SHIP MATERIALS ENGINEERING DEPARTMENT

DTRC ISSUES THREE TYPES OF REPORTS:

1. **DTRC reports, a formal series**, contain information of permanent technical value. They carry a consecutive numerical identification regardless of their classification or the originating department.

2. **Departmental reports, a semiformal series**, contain information of a preliminary, temporary, or proprietary nature or of limited interest or significance. They carry a departmental alphanumerical identification.

3. **Technical memoranda, an informal series**, contain technical documentation of limited use and interest. They are primarily working papers intended for internal use. They carry an identifying number which indicates their type and the numerical code of the originating department. Any distribution outside DTRC must be approved by the head of the originating department on a case-by-case basis.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
CRDKNSWC/HD-0936-06

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Carderock Division
Naval Surface Warfare Center
Ship Hydromechanics Dept

6b. OFFICE SYMBOL (If applicable)
Code 561

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and ZIP Code)
Bethesda, Maryland 20084-5000

7b. ADDRESS (City, State, and ZIP Code)

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Naval Undersea Warfare Center

8b. OFFICE SYMBOL (If applicable)
Code 3322

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)
New London, CT

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
63553N	S1704		DN1502204

11. TITLE (Include Security Classification)
USER'S MANUAL FOR THE SHIP MOTIONS PREDICTION APPLICATIONS MANAGER - PREDICT

12. PERSONAL AUTHOR(S)
T.C. Smith, C. Bennett

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM 05/93 TO 10/93

14. DATE OF REPORT (Year, Month, Day)
1994 Feb

15. PAGE COUNT
255

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Standard Ship Motions Program (SMP), Seakeeping
Time History
Application Manager

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report documents the use of a ship motions prediction applications manager (PREDICT) for use on a personal computer. PREDICT can generate both frequency and time domain output from a single ship input file. Hardcopy time history plots are also available. This report provides explanations of menu choices and examples of typical usage. The prediction programs used by PREDICT, the Standard Ship Motion Program (SMP), Simulation Time History Program (STH), and Access Time History Program (ACTH), are fully documented in the given references. This report also documents the personal computer version of the Standard Ship Motion Program (SMP93-PC).

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Timothy C. Smith

22b. TELEPHONE (Include Area Code)
301-227-5117

22c. OFFICE SYMBOL
Code 561

Previous editions are obsolete

S/N 0102-LF-014-6603

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

CONTENTS

FIGURES

TABLES

ABSTRACT

This report documents the use of a ship motions prediction applications manager (PREDICT) for use on a personal computer. PREDICT can generate both frequency and time domain output from a single ship input file. Hardcopy time history plots are also available. This report provides explanations of menu choices and examples of typical usage. The prediction programs used by PREDICT, the Standard Ship Motion Program (SMP), Simulation Time History Program (STH) and Access Time History Program (ACTH), are fully documented in the given references. This report also documents the personal computer version of the Standard Ship Motion Program (SMP93-PC).

ADMINISTRATIVE INFORMATION

This investigation was sponsored by the Naval Undersea Warfare Center, New London Laboratory, under Program Element 63553. The work was performed at the Naval Surface Warfare Center, Carderock Division, during FY1993 under work unit number 1-1541-802. The DN number is DN502204.

INTRODUCTION

The report describes a personal computer-based ship motion prediction applications manager, PREDICT, which provides both frequency and time domain predictions. PREDICT uses two sub-application managers to run the Standard Ship Motion Program(SMP93-PC)[1,2], the Simulation Time History Program(STH)[3], and the Access Time History Program(ACTH)[3] that make motion predictions. PREDICT also provides utilities to create and edit input files, and to view and plot output files.

The Standard Ship Motion Program (SMP) is a frequency domain motion prediction program. It calculates translational and angular ship responses in irregular seas for a range of ship headings, ship speeds, and modal wave periods at specific wave heights. The Standard Ship Motion Program uses Salvesen, Tuck, and Faltinsen strip theory[4] without end effect terms Heave, pitch, and surge are linear with respect to wave height. Roll, sway, and yaw are non-linear with respect to wave height and use an iterative calculation procedure with roll angle–dependent viscous damping. In this report, references to a specific SMP version will have the year attached, e.g. SMP84, whereas references to aspects common to all versions will be just SMP.

Both STH and ACTH are time domain motion prediction programs. The Simulation Time History program uses the SMP origin transfer function file to calculate ship time histories at the origin. The Access Time History program uses the origin time histories to generate time histories for any point on the ship. The time domain preserves the phase relation between motions which is lost in the frequency domain. As time goes to infinity, the time domain statistical values approach the frequency domain results.

Rather than repeating documentation for SMP, STH, and ACTH, this report deals exclusively with the PREDICT application managers, their menu choices, and organization. References 1, 2, and 3 deal with the theory behind SMP, STH, and ACTH. An example run is provided to show the typical path from SMP input file to ACTH time history output files.

Appendix A documents the main changes to SMP84 for the personal computer version of the Standard Ship Motion Program (SMP93-PC). Appendix B provides an overview of the applications managers in terms of directory structure and file location and function. Appendix C describes the ship specific input file for STH applications manager (STHAM). Appendix D is a listing of the source code for SMP93-PC.

This manual uses various typefaces to highlight important points and the relationships between sections. The `typewriter font` simulates a personal computer font and indicates DOS file and directory names, as well as, what the user sees on the screen, e.g. `BB62G8.HPL`. The **bold face** indicates manual section names, e.g. **INTRODUCTION**. The *italic face* indicates an important concept and, in file names, designates a wild card or variable. Main program names are in capital letters, e.g. SMPEDIT.

PREDICT OVERVIEW

PREDICT has three main branches: frequency domain, time domain, and data plotting. PREDICT uses SMP93-PC to make frequency domain predictions. Taking the frequency domain ship response transfer functions, the user with PREDICT runs STH and ACTH to make time domain ship motion predictions. Figure 1 shows the organizational structure and which programs are associated with which branch.

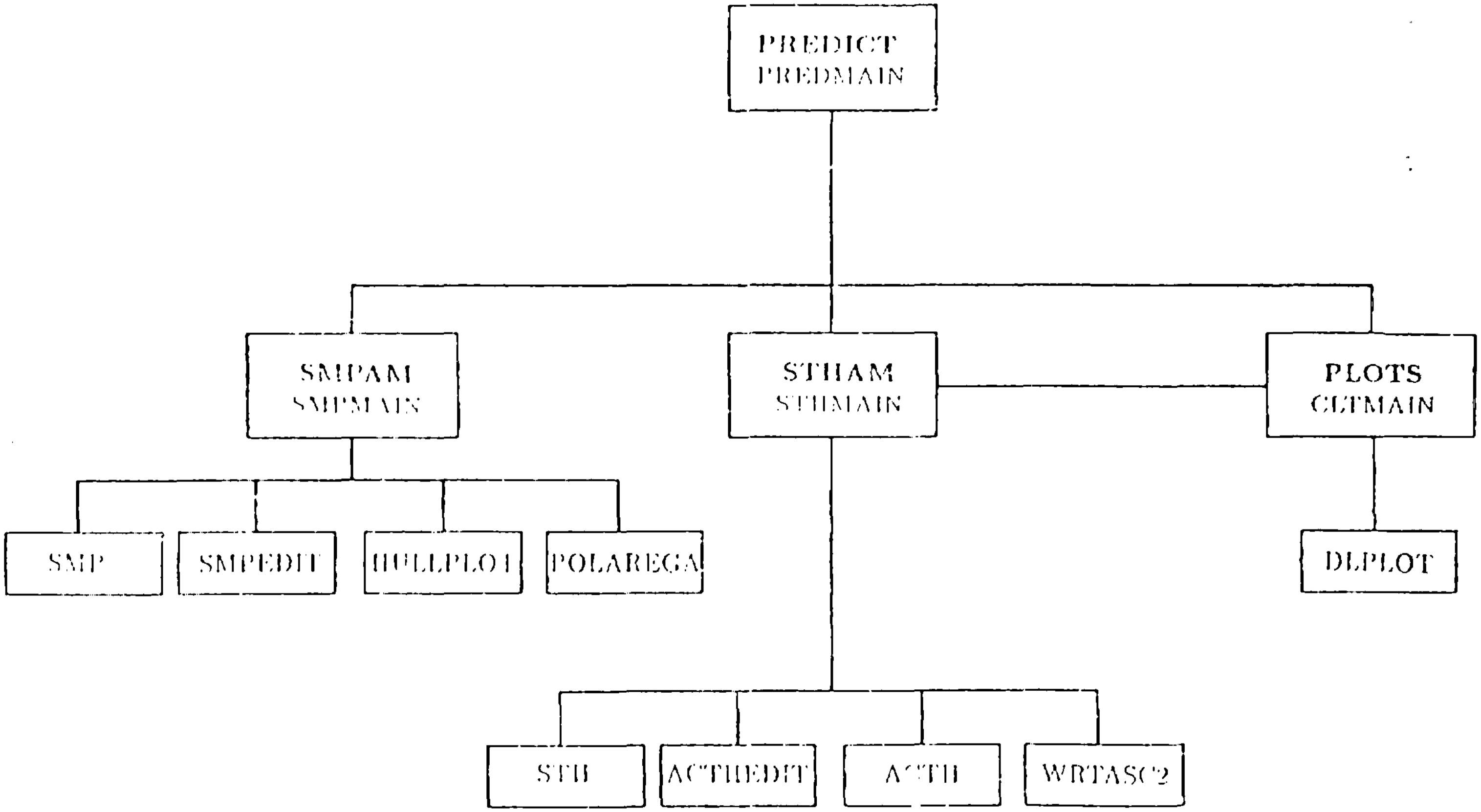


Fig. 1. PREDICT applications manager organizational structure.

PREDICT uses three sub-applications managers to run the three branches. The frequency domain sub-applications manager is SMPAM and it runs SMPMAIN.EXE which handles the menus described in **SMP APPLICATIONS MANAGER**, page 5. The time domain sub-applications manager is STHAM and it runs STHMAIN.EXE. **STH APPLICATIONS MANAGER**, page 16, describes the menu choices associated with STHAM. CLTAM is the plotting sub-applications manager and it runs CLTMAIN.EXE and DLPLOT.EXE. **Data Analysis and Plots**, page 25, deals with the menu choices for CLTAM.

Before running PREDICT, the user should set up a directory structure similar to the one described in Appendix B. The user needs to create executable and input directories and subdirectories, as well as install the commercial software required. PREDICT creates the output subdirectories as it needs them. The directories to create are in Table 1. The user created directories can be named any legal DOS name, though the examples in this manual use the default names. They can be on any accessible disk drive. The important item is that the executable directories have the right files in them: otherwise, PREDICT will not be able to find the programs to run.

Table 1. PREDICT directories to create and their function.

Option Name	Name	Description
SMP program path	SMP	SMPAM executables and help subdirectory.
SMP input path	SMPINPUT	SMP input files.
SMP output path	SMPOUTPT	SMP ASCII output files.
SMP data path	SMPDATA	SMP binary output files used by STHAM.
STH program path	STH	STHAM executables and help subdirectory.
STH data path	STHDATA	STH output subdirectories and files.
COLLECT data path	DATA	ACTH output subdirectories and files.
COLLECT program path	COLOCT91	CLTAM executables.

Each of the sub-applications managers has a data file which contains the directory paths for the different input and output. These files are: SMPSYS.TEX, STHSYS.TEX, and CLTSYS.TEX. The options for these files are explained under the SMP and STH descriptions. Though files can be changed from within PREDICT, mis-defining the run paths will cause PREDICT not to run. Appendix B describes in more detail the file directory structure and file locations.

The only input needed to start is an acceptable SMP93-PC input file. It should be located in a subdirectory of the SMP input directory. This subdirectory is referred to as a *Ship Type* when selecting input files, because typically ships of a given type are grouped together.

The input file name comprises a *root*, *variant*, *cycle number*, and an extension of `INP`. For example, DD965H6.INP is a representative input file of the DD963 class destroyer. The *root* is the main file designation and is typically the ship class and number, in this case, DD965. It has a maximum of five characters. The *variant* is a single letter that serves to distinguish major changes to a ship input file, e.g. different draft or loading conditions. The *cycle number* indicates the number of runs made with a particular input file. The example, `DD965H6.INP` has `DD965` as a root, `H` as a variant, 6 as the cycle number, and an extension of `INP`.

PREDICT names and creates all the output subdirectories automatically. Output file and directory names are based on the SMP input file name. SMP output files are split between output and data directories. In the output directory, there are *Ship Type* subdirectories and files have the root, variant and cycle of the input file. In the data directory, the files keep just the root and variant and are put in subdirectories named after the input file root. The STH output files are put in subdirectories that have SO added to the root and variant, e.g. `SODD965H`. The ACTH output files are put in subdirectories that have SP added to the root and variant, e.g. `SPDD965H`.

The example, page 28, details the process for generating frequency domain predictions, then time domain predictions, and finally plotting the data. It uses the default directory structure and assumes knowledge of it.

SMP APPLICATIONS MANAGER

The SMP Applications Manager (SMPAM) provides utilities that make running SMP and generating origin transfer function files easier. These utilities allow the selection of a ship input file, editing or viewing the input file, preliminary checking of the input file via a hydrostatics check and offset plots, running SMP, and viewing output files. There is also online help on the use of SMPAM.

SMPAM MAIN MENU OPTIONS

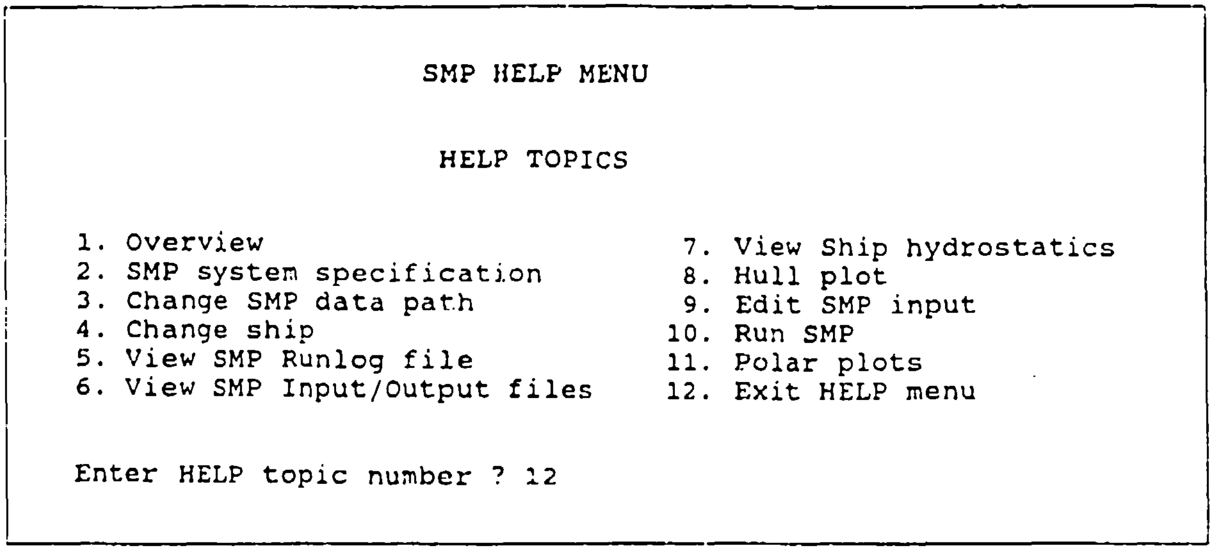
This section describes the main menu options of the SMP Applications Manager (SMPAM). Details of SMP input format and theory is in References 1 and 2. See Table 2 for a listing and brief description of the main menu selections.

Table 2 SMP Applications Manager main menu options and description

Number	Selection	Description
1	HELP	Online help for SMP Applications Manager. See Figure 2 for HELP options.
2	SMP System Specification	Sets default paths and directories for locating data for SMP runs.
3	Change SMP data path	Changes SMP input path without exercising selection 2.
4	Change ship	Changes ship directory, variant, and cycle without exercising selection 2.
5	View SMP runlog file	Pages through SMP runlog file, `SMPLOG.TEX`.
6	View SMP output file	Uses Norton^{TM} Editor[5] to view SMP output files.
7	View ship hydrostatics	Pages through hydrostatic calculations. It is necessary to run SMP first.
8	Hull plot	Plots the offsets or waterplane of the current ship. It is necessary to run SMP first.
9	View/Edit SMP input	Pages through input file or edits SMP input files using SMPEDIT.
10	Run SMP	Runs SMP and saves selected output.
11	Polar Plots	Generates polar and density plots.
12	Quit	Ends program; returns control to PREDICT.

HELP

HELP describes the various SMPAM main menu options. The HELP submenu, see Figure 2, is almost exactly like the SMPAM main menu it describes. This feature is similar to this manual, though it touches on slightly different aspects, e.g. running SMP in background, and lacks the overview of this manual.



```
                              SMP HELP MENU


                               HELP TOPICS


   1. Overview                             7. View Ship hydrostatics
   2. SMP system specification             8. Hull plot
   3. Change SMP data path                 9. Edit SMP input
   4. Change ship                         10. Run SMP
   5. View SMP Runlog file                11. Polar plots
   6. View SMP Input/Output files         12. Exit HELP menu


   Enter HELP topic number ? 12
```

Fig. 2. SMP applications manager HELP menu.

SMP System Specifications

The system specifications option can create, view, or modify the paths and directory specifications for SMP input and output. These specifications are in the file, SMPSYS.TEX, and could be created, viewed, or modified outside PREDICT.

This option has two submenus. The first specifies the action to take on the system specifications: either create, view, or modify. Choosing **Exit** at this point returns PREDICT to the SMPAM main menu. Choosing **Create** steps through all the specifications one at a time, asking the user to enter the appropriate path or directory. When complete, the data are written to SMP\SMPSYS.TEX.

Choosing to **View** the system specifications simply writes them to the screen and does not change them.

Choosing to **Modify** the system specifications allows the user to change just the specifications desired. The most typical changes, the SMP data path and ship variables, have their own options on the SMPAM main menu to make their use easier (Table 2). See Table 3 for a listing of the system specification options and a brief description.

Table 3. List and description of SMP system specifications menu.

Number	Selection	Description
1	Help	On-line help to explain menu options.
2	Halo^{TM} program path	Location of Halo^{TM} program files.
3	Halo^{TM} graphics screen driver	Screen driver identifier.
4	Halo^{TM} printer driver	Printer driver identifier.
5	SMP program path	Location of SMPAM executables.
6	SMP input path	Location of SMP input files.
7	SMP output path	Directory of SMP ASCII output.
8	SMP data path	Directory of SMP binary output.
9	Ship type	Subdirectory of SMP input directory.
10	Current ship	Root of ship name.
11	Variant	Denotes changes in hull geometry (A-Z).
12	Cycle	Number of SMP runs made with this variant.
13	Title	Default value is first line from current SMP input file.
14	Option	SMP run option.
15	Exit this menu	Returns to first system specification menu.

Appendix B explains the directory tree structure and file naming conventions used by PREDICT, SMPAM, and STHAM. It is best to read this appendix and understand the system specifications before making any changes to them.

The specification option, `Ship Type`, assumes that ships of a single type are put in the same directory. While this makes sense, it is not mandatory. It is also possible to specify a ship which does not exist, in which case very few options will work, though PREDICT will not stop running.

Changing the `Title` or `Option` here does not change the SMP input file.

Change SMP data path

The user can change the SMP binary output data location directly using this option, avoiding the menus associated with SMPAM main menu option 2.

The program displays the current path and asks the user if a change is desired. If so, the user enters the complete new path, starting at the root directory. If the path is correct as stated, the user enters `N`, indicating no further changes, and the program returns to the SMPAM main menu.

Change ship

Here, the user can choose from a list of available SMP input files, rather than having to remember them once running under PREDICT. The first step is to select the proper subdirectory from the directory specified in the SMP data path. The user enters the number of the desired subdirectory. Entering an `A` will accept the current highlighted choice.

The second step is to select the ship input file from that subdirectory. Again the user enters the number of the desired file or an `A` to accept the current choice. Only files with an `INP` extension are listed. An incorrect choice will result in an error warning and another chance to answer correctly.

View SMP runlog

This option pages through the SMP runlog file on the terminal screen. The SMP runlog file keeps track of the runs made with different variants, and displays the variant, cycle, title and comment from SMP93-PC input file. The run log file has the name *Root Variant*`.TLT` in directory `SMPOUTPT\`*Ship Type*, e.g. `SMPOUTPT\DESTROYR\DD965H.TLT`.

View SMP output files

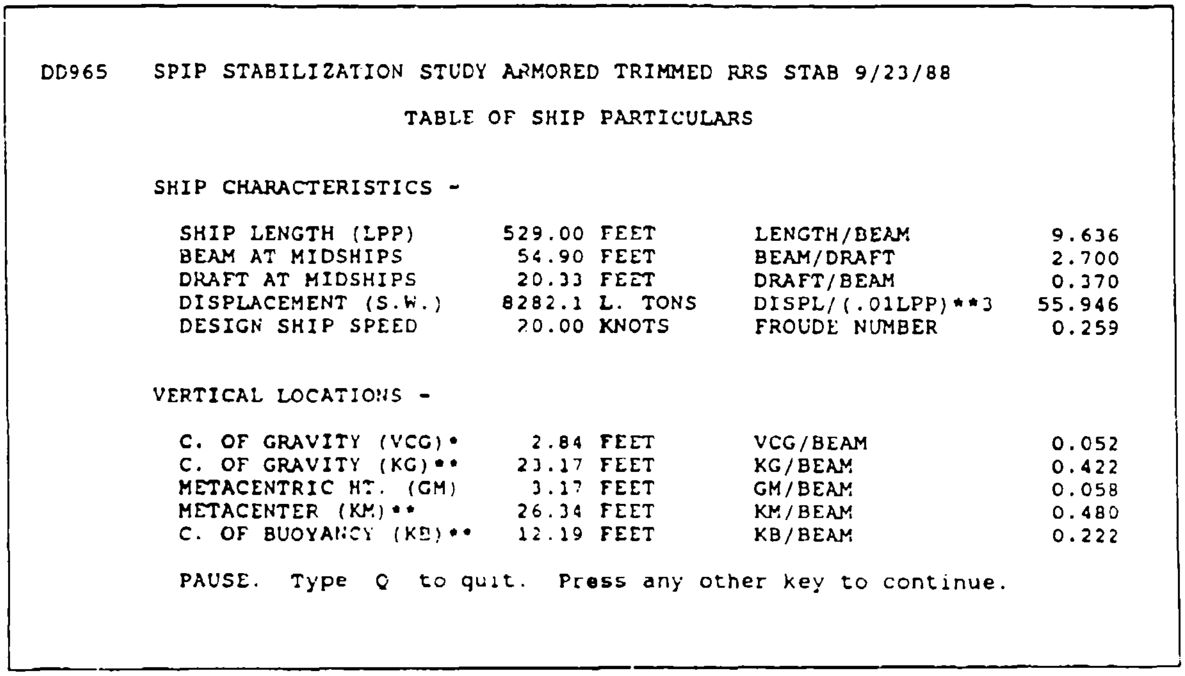
Under this option, it is only possible to view the files, not make or save any changes. The two selections on the submenu are shown in Table 4. PREDICT uses Norton™ Editor to view the output file. Once in Norton™ Editor, the standard Editor commands are valid though it is still not possible to save changes.

Table 4. View SMP output files sub-menu choices and descriptions.

Number	Selection	Description
1	View SMP output file using Norton^{TM} Editor	Uses BAT file to open Norton^{TM} Editor, no changes are saved.
2	Exit	Returns to SMPAM main menu.

View ship hydrostatics

This option pages through SMP hydrostatic data on the screen. See Figure 3, for an example of the first screen of data. The hydrostatic data include length, beam, draft, displacement, vertical and/or longitudinal centers of gravity, buoyancy, and flotation, various coefficients, and appendage data. It is necessary to have already run SMP for the current ship to use this option without an error message. The hydrostatic data file is HSTAT.TEX in directory SMPOUTPT\Ship Type, e.g. SMPOUTPT\DESTROYR\HSTAT.TEX.



```
DD965    SPIP STABILIZATION STUDY ARMORED TRIMMED RRS STAB 9/23/88

                          TABLE OF SHIP PARTICULARS

         SHIP CHARACTERISTICS -

           SHIP LENGTH (LPP)      529.00 FEET        LENGTH/BEAM             9.636
           BEAM AT MIDSHIPS        54.90 FEET        BEAM/DRAFT              2.700
           DRAFT AT MIDSHIPS       20.33 FEET        DRAFT/BEAM              0.370
           DISPLACEMENT (S.W.)    8282.1 L. TONS     DISPL/(.01LPP)**3      55.946
           DESIGN SHIP SPEED       20.00 KNOTS       FROUDE NUMBER           0.259

         VERTICAL LOCATIONS -

           C. OF GRAVITY (VCG)*     2.84 FEET        VCG/BEAM                0.052
           C. OF GRAVITY (KG)**    23.17 FEET        KG/BEAM                 0.422
           METACENTRIC HT. (GM)     3.17 FEET        GM/BEAM                 0.058
           METACENTER (KM)**       26.34 FEET        KM/BEAM                 0.480
           C. OF BUOYANCY (KB)**   12.19 FEET        KB/BEAM                 0.222

           PAUSE.  Type  Q  to quit.  Press any other key to continue.
```

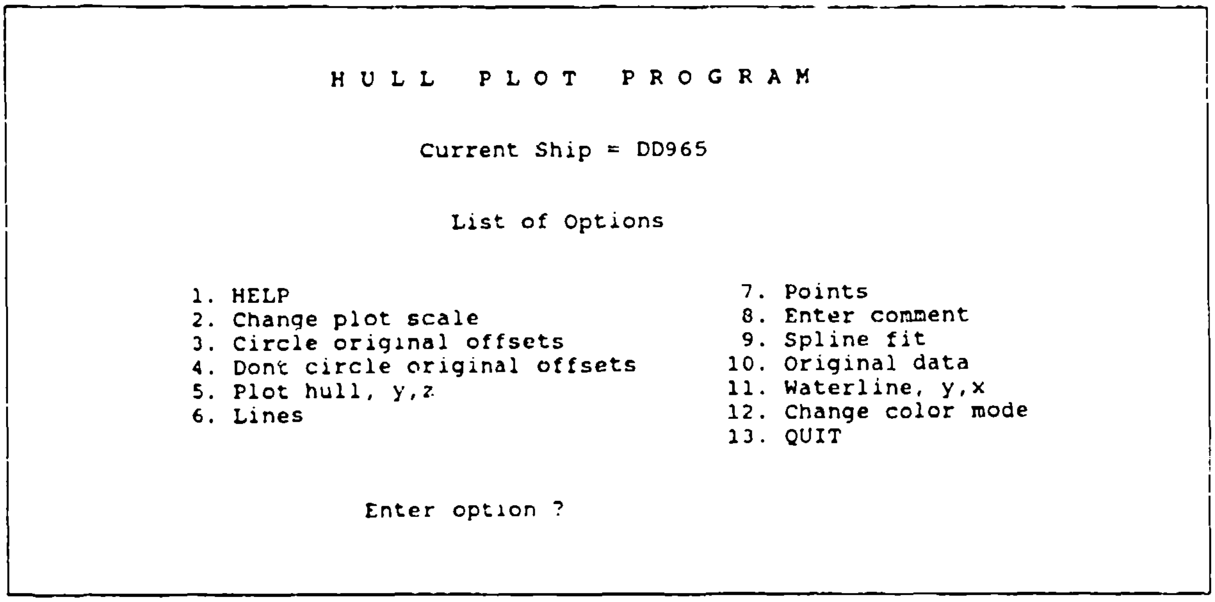
Fig. 3. First screen of SMP hydrostatics data.

Hull plot

HULLPLOT plots the offsets of the current ship using Halo™ Professional graphics package[6]. Displays of both body plan and waterplane plots are possible, as well as hard copies. The user can, to some extent, customize the appearance of the plots using the HULLPLOT submenu. It is necessary to first run SMP and save the *.HPL file for the current ship tc use this option. The HPL file is in directory SMPDATA, subdirectory *Root*, e.g. SMPDATA\DD965\DD965H6.HPL.

When running HULLPLOT, the user is queried as to whether the computer/printer is monochrome or color. Answering this question draws the body plan with default setting on the screen. The default settings result in a solid line plot of the spline fit without the original offset points shown, and the plot scale equal the maximum draft.

Entering H produces a hard copy; a carriage return brings up the HULLPLOT submenu, see Figure 4. The selections for this menu are in Table 5.



```
                         H U L L   P L O T   P R O G R A M

                                Current Ship = DD965

                                  List of Options

        1. HELP                                    7. Points
        2. Change plot scale                       8. Enter comment
        3. Circle original offsets                 9. Spline fit
        4. Dont circle original offsets           10. Original data
        5. Plot hull, y,z                         11. Waterline, y,x
        6. Lines                                  12. Change color mode
                                                  13. QUIT

                          Enter option ?
```

Fig. 4. HULLPLOT submenu choices.

Plot settings are turned on by choosing that selection and remain in effect until counteracted by another option. Settings that are not mutually exclusive can be combined to produce the desired look.

Table 5. HULLPLOT plot features menu listing and description.

Number	Selection	Description
1	HELP	Help feature that explains the HULLPLOT program.
2	Change plot scale	Changes plot size. Value specified is plot height; the width is autoscaled to avoid scaling distortion.
3	Circle original offsets	Draws circles around original offsets.
4	Don't circle original offsets	Removes circles around original offsets.
5	Plot hull, y,z	Plots body plan using current settings.
6	Lines	Uses solid lines for drawing.
7	Points	Uses dotted lines for drawing.
8	Enter comment	Allows additional comments to be added to hard copy plots. Plots already use input file title as plot title.
9	Spline fit	Plots spline fit of original offsets.
10	Original data	Plots lines through original offsets. Using Points (7) with this option results in just the offset data with no lines.
11	Waterline, y,x	Plots ship waterplane.
12	Change color mode	Changes display to and from monochrome to color.
13	QUIT	Returns to SMPAM main menu.

Edit SMP input

SMPEDIT is a guided editor for SMP input files. SMPEDIT allows users who are not overly familiar with the input file format to make changes while reducing the risk of entering an error. The main menu is virtually a listing of the Data Card Set names from Reference 1.

The process of making changes is fairly straightforward and demonstrated fully in the example, page 28. The user selects a Data Card Set to modify. The next submenu is either a listing of the Data Card Set variables to choose from, or a submenu asking what action is to be taken, i.e., deleting, creating, or modifying

In the first case, the user simply selects the variable to change and enters the new value. When through, the user selects EXIT and returns to the SMPEDIT main menu.

In the second case, the user enters what action to take, typically deleting, creating, modifying, or exiting. This second type of submenu occurs with hull geometry, appendage, point location, and sea state options. A series of questions follow that guide the user to make the desired changes by displaying the current values and prompting the user for new ones. When through, the user selects EXIT and returns to the SMPEDIT main menu.

Option 15, Edit Annotation, is not an official Data Card Set from Reference 1 or 2. It is additional identifying information added to the SMP93-PC input file and echoed in the SMP runlog. It does not appear in the SMP output and does not affect computations.

The last option, Write changes to SMP input file and exit, option number 16, requires the user to understand the naming convention using variants and cycles. Changes to the input file hull geometry will cause the variant to change, but appendage changes alone will not change the variant. The "save" submenu gives two possibilities for saving the file.

The first choice will always create a new file with a new cycle number. If the variant changed, then the new file will also have a new variant. For example, the modified file, DD965H6, is saved as either DD965H7 or DD965I7, depending if the variant changed.

The second choice, will overwrite the existing file if the variant did not change. If the variant did change, the new file will have a different variant but same cycle number. For example, the modified file, DD965H6, is saved as either DD965H6 or DD965I6, depending if the variant changed.

It is important to keep track of which variant and cycle have what modification to know what is in which subdirectory. The annotation line is useful for this.

Run SMP

Runs SMP93-PC using the current ship and directs output to the proper directories. The Run SMP menu allows the user to decide which files to keep and which to delete after running SMP. Not all these files are created for every SMP run. Which files are generated by SMP depends on the output options flagged in the Program (Run) Options line of the SMP input file. Reference 1 and SMPEDIT describe run options and how they affect the output. Choosing to save a file does not generate the file if those output options are not chosen in the SMP input file. The selections simply toggle between yes and no. The possible files are listed in Table 6.

Table 6. Description of possible SMP output files.

Menu choice	Description	Type †	Extension
Potential file	Potential flow velocity potential	B	POT
Coefficient file	Added mass and damping, excitation	B	COF
Load coefficient file	Loads	B	LCO
Hull plot file *	Spline fit of offsets for HULLPLOT	A	HPL
Load response operator file	Response operators for loads	B	LRA
Origin file *	Ship origin transfer function file	B	ORG
Response operator file	Response amplitude operators	B	RAO
RMS file	Response RMS for unit wave height	B	RMS
Severe motion file	Worst case response and sea conditions	B	SEV
Speed polar data file *	Response data for speed polar plots	B	SPD
Speed polar text file *	Labels and titles for speed polar plots	A	SPT
Lateral coefficient file	Frequency domain coefficients for rudder roll stabilization	B	LAC
Lateral excitation file	Frequency domain coefficients for rudder roll stabilization	B	LAE

* saving recommended

† A=ASCII; B=Binary

Polar plots

The polar plot routine, POLAREGA, provides two types of plots: a polar plot of the response and a density plot of the response versus encountered modal period. The polar plots can either be color filled contours or black and white. It is necessary to run SMP and save the SPD and SPT files before attempting polar plots. If PREDICT cannot find the SPD and SPT files for the current ship, it simply returns to the main menu.

Like the rest of PREDICT, POLAREGA is menu driven. Table 7 gives the options on the polar plot main menu and Table 8 gives the options on the density plot main menu. POLAREGA uses values from the previous plot as default values for the current plot.

Table 7. Polar plot main menu options.

Number	Selection
1	Select Channel
2	Change Sea Condition
3	Select Magnitude or Period
4	Select Color or Black and White Print
5	Change Scale
6	Plot Polar Plot
7	Exit

Table 8. Density plot main menu options.

Number	Selection
1	Select Channel
2	Change Scales
3	Change Sea Type
4	Plot Density
5	Exit

A *channel* is simply a ship response. The ship response name is the name used internally to SMP to identify responses. In that naming convention: displacement is DSP; velocity is VEL; acceleration is ACC; lateral is LAT; longitudinal is LON; vertical is VER; relative motion is RLM, relative velocity is RLV; and points are P1 to P0. So pitch velocity would be PITVEL and vertical acceleration of point 3 would be VERACCP3.

Selecting a response also gives the user a chance to set the wave direction and change the scales.

The *sea condition* is a 10 character string, beginning with the Bretschneider designation "BR", containing the significant wave height, modal period, and crestedness of the seas. Again POLAREGA uses the SMP internal representation. The first four digits are the significant wave height without a decimal place. The next two digits are the modal period. The last two characters denote either longcrested or shortcrested. For example, BR061911SC would be a shortcrested Bretschneider spectra with a significant wave height of 6.19 feet and modal period of 11 seconds. Changing the sea condition also gives the user a chance to set the wave direction.

Changing the *scale* of a polar plot simply determines the maximum and minimum contours and the contour increment. On the density plot, changing the scale affects the y axis only.

Choosing between *magnitude* or *period* is a choice between plotting the magnitude or the encountered modal period of the response. Selecting a response also gives the user a chance to set the wave direction and change the scales.

Setting the wave direction equal to 0 means the ship's bow is pointed towards 0 degrees. This is the most common setting. Using a non-zero value is useful for comparing with full scale trials data where heading has actual geographical significance. The speeds for the plot come from the SMP input file.

To generate a polar plot, step through the options selecting a response, sea condition, black and white or color, and response or period. There are many opportunities to change the scale, so it is not always necessary to do so. POLAREGA reads SMPSYS.TEX, POLAR.DAT, and DENSITY.DAT for the path and previous run data.

Quit

Returns user to PREDICT main menu.

STH APPLICATIONS MANAGER

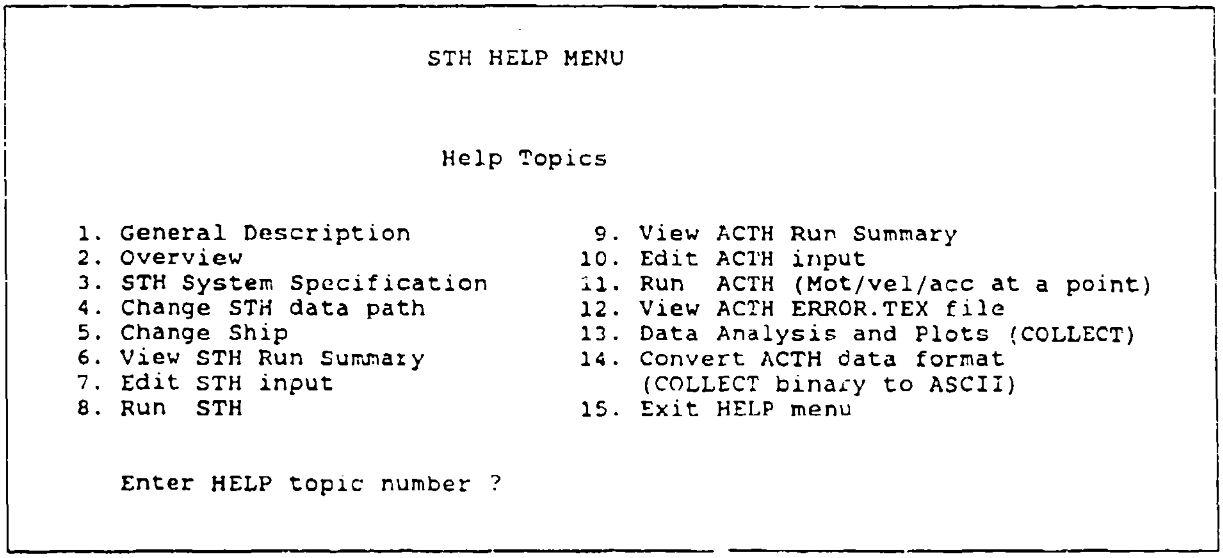
The Simulation Time History Applications Manager (STHAM) helps the user in setting up STH and ACTH runs. The options allow the user to set data paths, select ships, create STH and ACTH input files, run STH and ACTH, and plot the time histories.

STHAM MAIN MENU OPTIONS

This section is a brief description of the main menu options and what they do. Details as to input file formats and program theory are in Reference 3. See Table 9 for listing of main menu options and a brief description.

HELP

HELP gives a description of the general overview of the STH applications manager and the main menu options. The HELP menu, see Figure 5, is very similar to the STHAM main menu and provides the same sort of help as the SMPAM online help.



```
                              STH HELP MENU


                               Help Topics

   1. General Description             9. View ACTH Run Summary
   2. Overview                       10. Edit ACTH input
   3. STH System Specification       11. Run  ACTH (Mot/vel/acc at a point)
   4. Change STH data path           12. View ACTH ERROR.TEX file
   5. Change Ship                    13. Data Analysis and Plots (COLLECT)
   6. View STH Run Summary           14. Convert ACTH data format
   7. Edit STH input                     (COLLECT binary to ASCII)
   8. Run  STH                       15. Exit HELP menu

      Enter HELP topic number ?
```

Fig. 5. STH applications manager HELP menu.

STH System Specifications

The system specifications option creates, views, or modifies the paths and directory specifications for time history input and output. These specifications are in the file, STHSYS.TEX, and could be created, viewed, or modified outside PREDICT.

This option has two submenus. The first specifies the action to take on the sys-

Table 9. STH Applications Manager main menu selections and description.

Number	Selection	Description
1	HELP	Online help for STHAM.
2	STH System Specifications	Sets default paths and directories for locating data for STH and ACTH runs.
3	Change STH data path	Changes STH input path without exercising selection 2.
4	Change Ship	Changes ship directory, variant, and cycle with exercising selection 2.
5	View STH Run Summary	Pages through STH run summary file, STHLOG.TEX.
6	Edit STH input	Accesses STH input file so user can set proper sea conditions.
7	Run STH	Runs STH for current ship and input files.
8	View ACTH Run Summary	Page through ACTH run summary file, ACTHLOG.TEX.
9	Edit ACTH input	Accesses ACTH input file so user can select point locations for time histories.
10	Run ACTH (Mot/vel/acc at a point)	Runs ACTH for current ship and input files.
11	View ACTH ERROR.TEX file	Pages through ACTH run time error file.
12	Data Plot	Plotting routines for hardcopy output of STH and ACTH time histories.
13	Convert ACTH data format	Converts binary output files to ASCII.
14	QUIT	Returns to PREDICT main menu.

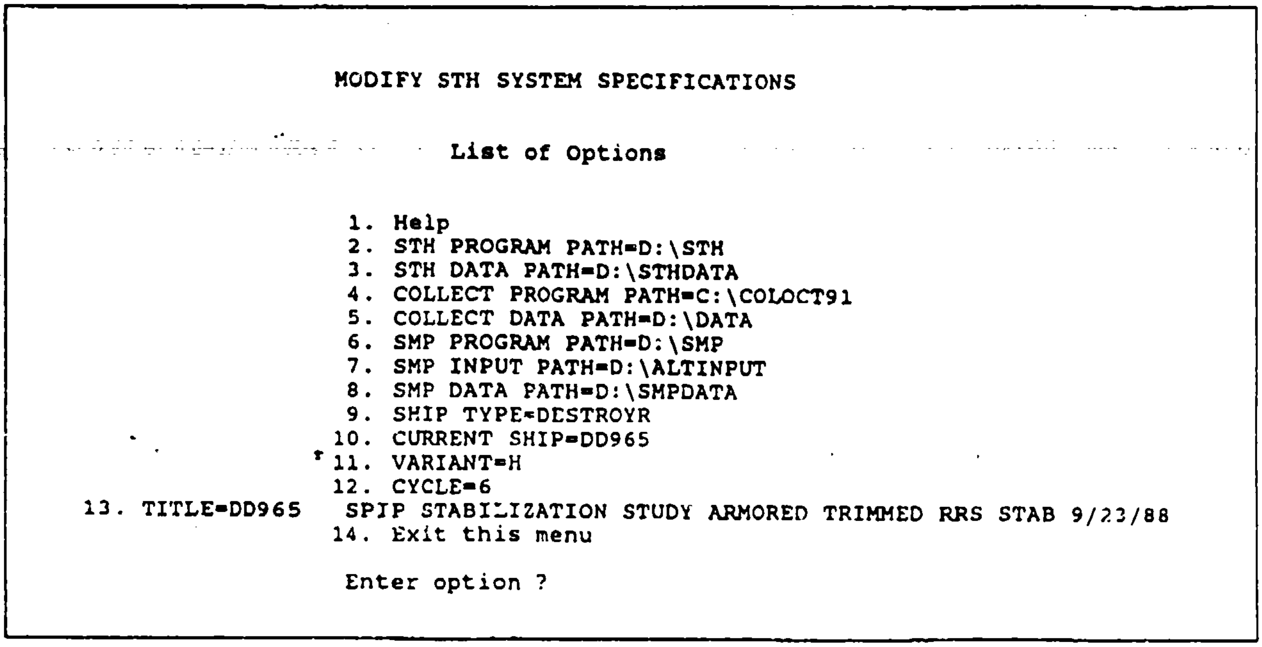
tem specifications, either create, view, or modify. Choosing **Exit** at this point returns PREDICT to the STHAM main menu. Choosing **Create** steps through all the specifications one at a time, asking the user to enter the appropriate path or directory. When complete the data are written to STHSYS.TEX.

Choosing to **View** the system specifications simply displays them on the screen and does not change them.

Choosing to **Modify** the system specifications brings up the second submenu, see Figure 6. Using the second submenu, the user can change just the specification desired. The most typical changes, the STH data path and ship variables, can be made more easily using options 3 and 4 of the STHAM main menu, Table 9. Again Appendix B gives a description of the file directory structure and default paths. Table 10 has a listing and brief description of the system specification menu options.

Table 10. STH system specifications listing and description.

Number	Option	Description
1	Help	On-line help to explain menu options.
2	STH program path	Location of STHAM executables.
3	STH data path	Location of STH output files.
4	COLLECT program path	Location of COLLECT executables.
5	COLLECT data path	Location of ACTH output.
6	SMP program path	Location of SMPAM executables.
7	SMP input path	Location of SMP input files.
8	SMP data path	Location of ship origin transfer function files, usually the SMP binary output directory.
9	Ship type	Subdirectory of SMP input directory.
10	Current ship	Root of ship name.
11	Variant	Denotes changes in hull geometry. (A-Z)
12	Cycle	Number of SMP runs made with this variant.
13	Title	Default value is first line from current SMP input file.
14	Exit this menu	Returns to first STH system specification menu.



```
                     MODIFY STH SYSTEM SPECIFICATIONS

                              List of Options

                       1. Help
                       2. STH PROGRAM PATH=D:\STH
                       3. STH DATA PATH=D:\STHDATA
                       4. COLLECT PROGRAM PATH=C:\COLOCT91
                       5. COLLECT DATA PATH=D:\DATA
                       6. SMP PROGRAM PATH=D:\SMP
                       7. SMP INPUT PATH=D:\ALTINPUT
                       8. SMP DATA PATH=D:\SMPDATA
                       9. SHIP TYPE=DESTROYR
                      10. CURRENT SHIP=DD965
                      11. VARIANT=H
                      12. CYCLE=6
      13. TITLE=DD965   SPIP STABILIZATION STUDY ARMORED TRIMMED RRS STAB 9/23/88
                      14. Exit this menu

                       Enter option ?
```

Fig. 6. STH system specifications menu.

Change STH data path

Here the user can only change the drive of the STH data path. The directory name remains unchanged. The drive and directory name can be changed using STHAM main menu option 2, `STH Systems Specifications`.

Change ship

With this option, the user selects a new ship from a displayed list of available SMP origin transfer function files, and STH input files. First the user can choose from existing ship time history input files located in `STH` by entering the new ship's number, or enter `A` to accept the current ship.

If the desired ship does not yet have a time history input file, enter `N`. Whereupon, the user selects a *Ship Type* subdirectory and ship from the SMP input directory, in the manner described for selecting ships with the SMPAM.

View STH Run Summary

This option displays the STH run summary file, STHLOG.TEX, on the terminal screen. The STH run summary keeps track of which origin file was used with what sea conditions, i.e. significant wave height, wave modal period, ship speed and heading. Each ship has its own run summary file located in STHDATA\SO*Root Variant*, e.g. STHDATA\SODD965H.

Edit STH input

STHEDIT allows the user to change the STH input in an orderly fashion that minimizes input error. STHAM uses two input files: a ship dependent input file that also has ACTH data, and a generic input file, STH.INP. STHEDIT updates both these files. Appendix C describes the ship specific input file. Reference 3, pages 16–20, describes the STH input file, STH.INP, format fully. Once the user becomes familiar with the input format, it may be easier and faster to edit the files manually by accepting the current selections and then opting to edit the file using Norton^{TM} Editor later.

STHEDIT presents all the STH relevant data on one screen, and *the user changes the input by entering the number of the data and its new value, separated by a comma. The new value should include the decimal point.* When finished, the user then selects the starting run numbers and may edit the input file to remove unwanted sea conditions. See the example in the **STHEDIT**, page 37, for information about choosing sea conditions, run numbers, and changes to make using Norton^{TM} Editor.

Run STH

Runs STH using the current ship origin transfer function file and STH input file. Output files SR*N*.TEX and SR*N*.DAT where *N* is the run number, are written to SO*Root Variant* subdirectory in the directory specified by STH DATA path.

View ACTH Run Summary

This option pages the ACTH run summary file, TRIALLOG.TEX, onto the terminal screen. The ACTH run summary keeps track of which sea conditions, i.e. significant wave height, wave modal period, ship speed and heading, were used for a run. Each ship has its own run summary file located in DATA\SP*root variant*, e.g. DATA\SPDD965H.

Edit ACTH input

ACTHEDIT allows the user to change the ACTH input file. STHAM uses two input files to run ACTH: a ship specific input file that also has STH data, and a generic ACTH input file, ACTH.INP. Appendix C describes the ship specific input file. Reference 3, pages 22–27, describes the ACTH input file format fully. It is always possible to edit input files using some other editor, and may be faster and easier once the user is familiar with the input format.

Table 11. ACTHEDIT main menu options and description.

Number	Option	Description
1	HELP	Provides online help about ACTH input file
2	Edit Wave Point Locations	Delete, create, or modify wave point locations. Must specify same locations as STH run, or ACTH will not run.
3	Edit Point Locations	Delete, create, or modify points at which to calculate absolute motion.
4	Edit channels	Specify responses and points for time history calculations.
5	Select STH Runs	Chose STH run, or trial, to use as basis for ACTH time histories by flagging or unflagging ship and sea conditions.
6	Select OUTPUT format	Choose between either binary or ASCII output.
7	Exit this menu	Returns to STHAM main menu.

The organization of ACTHEDIT is like SMPEDIT, having menus and submenus, rather than STHEDIT which makes selections from just one list. Table 11 lists the ACTHEDIT main menu options.

The subscreens are all fairly similar in that they have submenus that allow the existing data to be modified or deleted, or new data added. It is simply a matter of making the desired choice and entering the data. See the example section **ACTHEDIT**, page 43, for information about modifying data and selecting STH runs.

Some choices do require some extra explanation:

Wave point locations. STHAM will not allow the user to enter wave point locations if the selected STH runs do not have wave points. So it necessary to specify the STH runs with wave points before editing the wave point locations. Also if the wave points locations are not identical between the STH runs and ACTH input, then ACTH will not run.

Edit Point Locations The actual mechanics of adding, deleting, or modifying point locations are simple: ***separate data with commas and include decimal points.*** But changing the point locations here does not update the point location data used elsewhere. It is necessary to modify the channels selected to make sure they are using the new point locations.

Channel selection. Channel selection involves menus which determine which response is saved in what channel. Only responses with assigned channels are saved in the output files. A channel is simply a column of numbers with data saved for every time step. A maximum of 16 channels are possible.

The first submenu asks whether to modify, delete, or add a channel. The next submenu asks for a channel number to operate on. And the final submenu lists the possible responses to assign to channels. Table 12 lists the type of responses possible and which submenu choice they are associated with. The final selection involves specifying whether a response is vertical, lateral, or longitudinal; a displacement, velocity, acceleration, or force; or is ship-referenced or earth-referenced.

ACTH uses channels and points from the previous run as initial values and does not update point locations automatically. If these are not appropriate, it is necessary to either modify the existing channels, or delete them and add new ones. When adding a new channel, wave height is always the default channel and is then changed by further selections.

Select STH runs. STHAM presents the conditions used to generate the STH output files in the specified STH data directory and ship subdirectory automatically.

Table 12. Possible response selections.

Selection	Description
Wave height origin	Wave elevation at origin
Origin motion	Six degree of freedom response - displacement, velocity, and acceleration
Motion at a point	Longitudinal, lateral, and vertical displacement, velocity, and acceleration of a point in Earth reference frame.
Relative Motion point	Longitudinal, lateral, and vertical displacement, velocity, and acceleration of a point in ship reference frame.
Wave height at a Point	Wave elevation at a point.
Forces at a point	Longitudinal, lateral, and vertical forces at a point in Earth or ship reference frame.

Thus, the user can only flag or unflag the choices in this matrix; it is not possible to increase the matrix of conditions here. If the desired condition does not exist among the list of choices, either specify a new STH data directory and ship subdirectory, or make a STH run for the desired conditions.

Run ACTH (Mot/vel/acc at a point)

Calculates motion at a point time histories by running ACTH using the current ship origin time history files and ACTH input file. Writes output files, `DR`*N*`.TEX`, `DR`*N*`.ASC`, `DR`*N*`.INT`, `DR`*N*`.CON`, where N is the run number, into the COLLECT data path directory and ship subdirectory, e.g. `DATA\SPDD965B`.

View ACTH ERROR.TEX file

Displays file, `ERROR.TEX`, to the screen. `ERROR.TEX` contains any error messages written while running ACTH. If ACTH ran without error, `ERROR.TEX` has the message: `All STH runs were successfully completed`.

Data Analysis and Plots

The Data Analysis and Plots option plots the ACTH time history data. The current maximum amount of data that can be plotted is 16 channels at 3 samples per second for 20 minutes.

The first submenu, titled **Datalog data analysis and plotting**, has options to select the data path and run to plot. Table 13 lists the menu options.

Table 13. Datalog data analysis and plotting menu options.

Number	Option	Description
1	Computer system specification	Modify COMPSYS.TEX
2	Change DATA Analysis drive	Changes drive of DATA directory.
3	Change TRIAL	Select new *TRIAL* subdirectory.
4	Get new run	Select new run from current DATA analysis path.
5	Plot data	Continues with plotting menu.

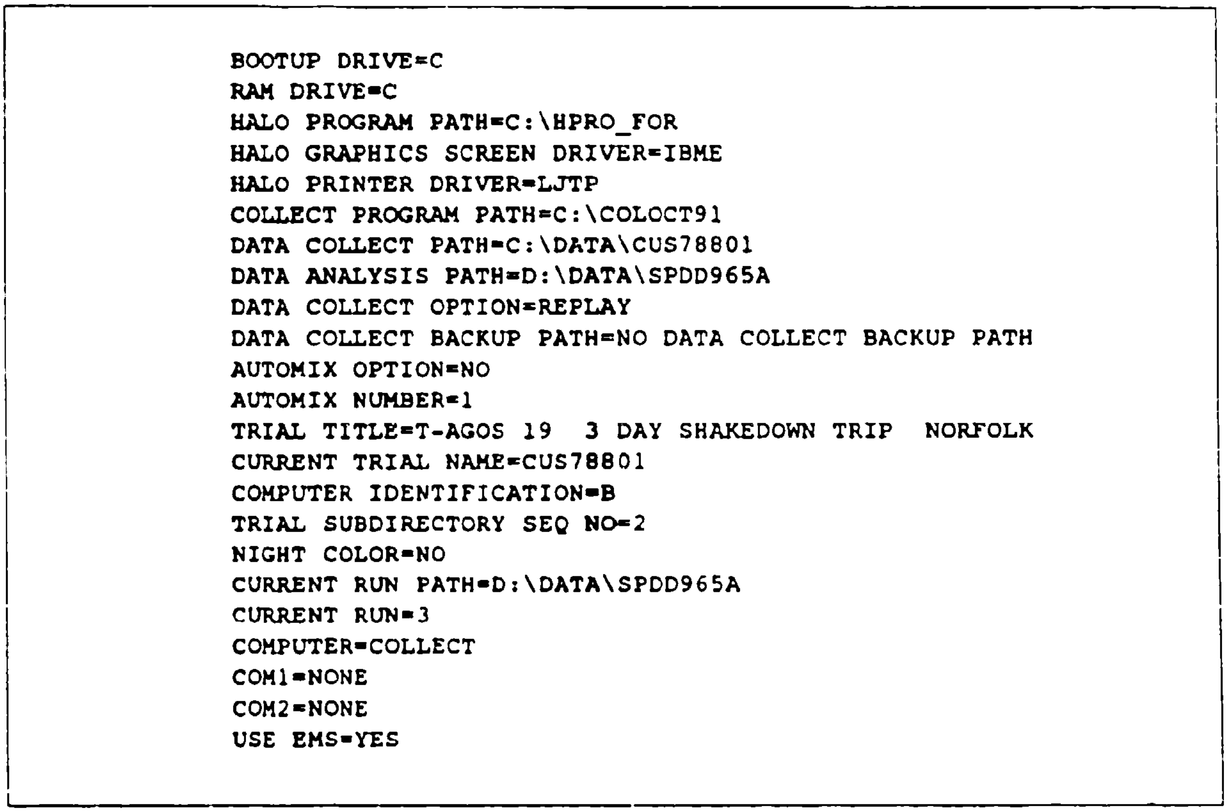
Changes to COMPSYS.TEX COMPSYS.TEX has path and file data for CLTMAIN, the way SMPSYS.TEX has data for SMPMAIN. See Figure 7 for an example COMPSYS.TEX file. PREDICT only uses CLTMAIN for plotting and most of the data in COMPSYS.TEX is extraneous for plotting purposes. Table 14 lists and briefly describes the lines pertinent for plotting.

Data path variables Options 2 - 4 work like the SMPAM and STHAM data path related options, except some of the terminology in new. *TRIAL* refers to the subdirectory names in DATA, the directory specified by COLLECT data path. Their names have the form SP*Root Variant* and are analogous to the subdirectories in STHDATA. A *run* refers to the set of sea conditions used when calculating the time history data. Run numbers are assigned to sea conditions in the same order as the in the ACTH input file.

Datalog plotting This menu has the details for determining the look of the time history plots. Here the user selects the channels and scales to use for the plots. It is also

Table 14. COMPSYS.TEX lines used by PREDICT.

Number	Option	Description
1	HELP	On-line help to explain menu options.
4	Halo^{TM} program path	Location of Halo^{TM} files.
5	Halo^{TM} graphics screen driver	Screen driver identifier.
6	Halo^{TM} printer driver	Printer driver identifier.
7	COLLECT program path	Location of CLTAM executables.
9	DATA analysis path	Directory of ACTH output.
19	Current run path	Directory of current run data.
20	Current run	Number of run currently plotting.
25	Exit	Returns to previous menu.



```
BOOTUP DRIVE=C
RAM DRIVE=C
HALO PROGRAM PATH=C:\HPRO_FOR
HALO GRAPHICS SCREEN DRIVER=IBME
HALO PRINTER DRIVER=LJTP
COLLECT PROGRAM PATH=C:\COLOCT91
DATA COLLECT PATH=C:\DATA\CUS78801
DATA ANALYSIS PATH=D:\DATA\SPDD965A
DATA COLLECT OPTION=REPLAY
DATA COLLECT BACKUP PATH=NO DATA COLLECT BACKUP PATH
AUTOMIX OPTION=NO
AUTOMIX NUMBER=1
TRIAL TITLE=T-AGOS 19  3 DAY SHAKEDOWN TRIP  NORFOLK
CURRENT TRIAL NAME=CUS78801
COMPUTER IDENTIFICATION=B
TRIAL SUBDIRECTORY SEQ NO=2
NIGHT COLOR=NO
CURRENT RUN PATH=D:\DATA\SPDD965A
CURRENT RUN=3
COMPUTER=COLLECT
COM1=NONE
COM2=NONE
USE EMS=YES
```

Fig. 7. COMPSYS.TEX example file.

possible to look at the actual data in a text format. Table 15 lists and briefly describes the menu options.

Changing the Y scale requires some further explanation. DLPLOT uses either default values of one, or the previous Y ranges for the current plot. There are three scaling options when changing the Y scale. Entering a single N autoscales all the channels. Entering a N,*channel number*, e.g. N,3, autoscales just that channel. To manually scale a channel, enter *channel number, maximum y value, minimum y value, y increment*, e.g. 2, 3.0, -2.0, 1.. Remember to separate data by commas and include the decimal point. Entering a zero exits this option.

Table 15. Datalog plotting submenu options.

Number	Option	Description
1	HELP	Online help about DATALOG plotting.
2	Get run	Choose different run number from current DATA analysis path.
3	Edit channels	Change channels to plot (6 maximum for QPLOT).
4	Graph type	Change type of graph.
5	Change Y scale	Overrides default scaling values for y axis.
6	Change X increment	Changes x axis tic mark increment.
7	Time	Changes starting and stopping time of plot.
8	Plot data	Plots data to screen.
9	Display data	Pages through the digital time history data for the selected channels.
10	Display text file	Pages through run log and minimum analysis of current run.
11	Exit	Returns to Data Analysis and Plotting Menu.

Convert ACTH data format (binary to ASCII)

Converts ACTH time history data from a binary format to an ASCII format for more general use. Reference 3, pages 28 - 29, describes both the binary and ASCII file formats.

Quit

Returns control to PREDICT main menu.

EXAMPLE PREDICT RUN

This section is a start to finish example of using PREDICT to generate motion at a point time histories from a SMP93-PC input file. Reference 3, pages 30–33, has an example of running STH and ACTH not using the application manager.

This example assumes the existence of the SMP93-PC input file. References 1 and 2 detail the correct SMP input file format. The system specifications are set to the default values given in Appendix B. The figures are the actual screens from the run. The user is encouraged to use this section as a tutorial and follow along on their computer. *To select an option, type that option's number at the prompt and press the* `RETURN` *key.*

EXAMPLE PROBLEM STATEMENT

The example generates motion–at–a-point time histories using an existing destroyer input file by running SMP, STH, and ACTH. The input file is `DD965H6.INP` and will be in subdirectory `ALTINPUT\DESTROYR`. We want motion–at–a-point time histories of a boom tip in two sea conditions. The boom tip location is station 20.1, 15 feet starboard of centerline, and 40 feet up from the baseline.

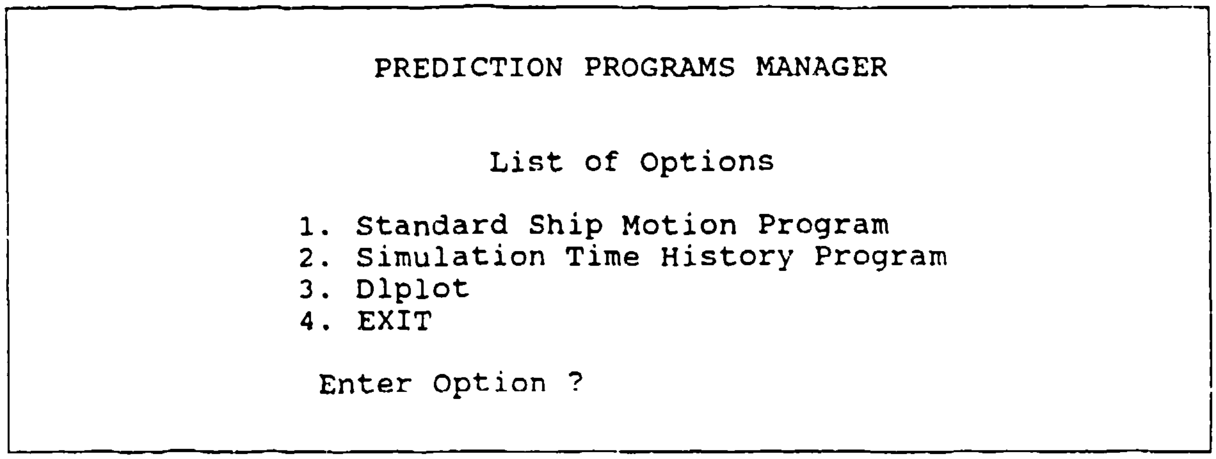
The two sea conditions are: 20 knots, head seas (0 degrees), 6.2–foot significant wave height, and 9–second modal period; and 10 knots, beam seas (90 degrees), 6.2–foot significant wave height, and 7–second modal period. The response to plot is the vertical acceleration of the boom tip.

PRE-RUN SET UP

The first step is create a properly formatted SMP input file before running PREDICT. If the input file is not correctly formatted, PREDICT will return to the DOS prompt when the user tries to run SMP or edit the file. The file name should follow the specifications in NAMING CONVENTIONS, page 5, and have the `INP` extension, otherwise PREDICT will not be able to find the file. Secondly, the user needs to know the directory where the input file is located so the path can be specified correctly. For the example, the input file, DD965H6.INP, will be in subdirectory ALTINPUT\DESTROYR.

RUNNING SMP

Having set up the input file and directory, the user types: PREDICT at the DOS prompt. The user sees Figure 8 on the screen, and enters a 1 to select Run SMP Application Manager.



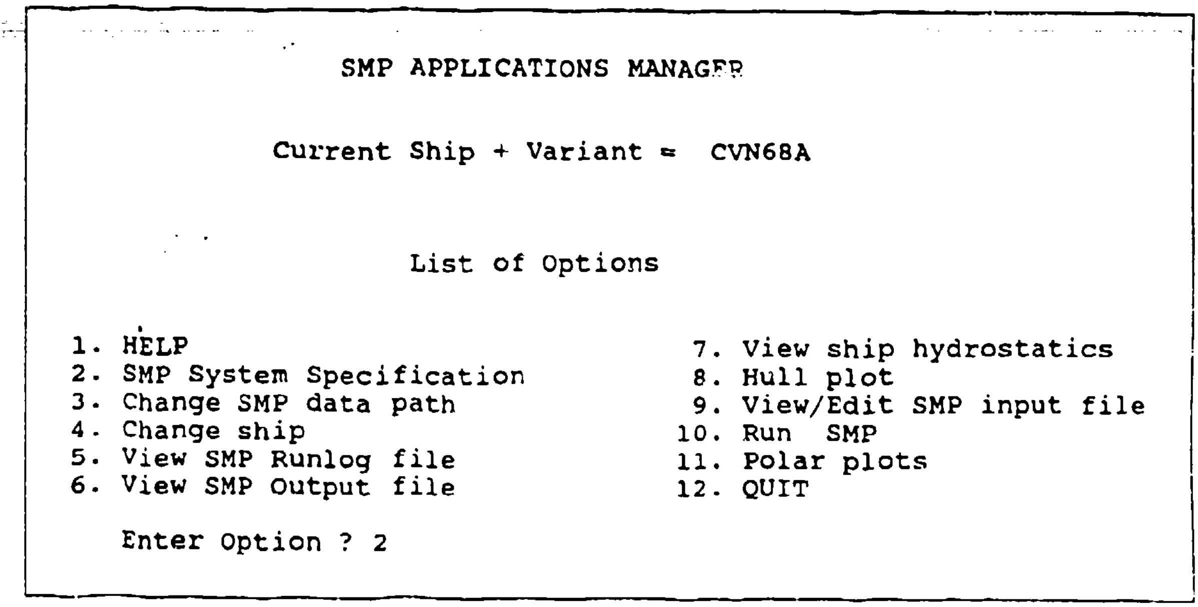
```
                 PREDICTION PROGRAMS MANAGER


                       List of Options

            1. Standard Ship Motion Program
            2. Simulation Time History Program
            3. Dlplot
            4. EXIT

             Enter Option ?
```

Fig. 8. PREDICT main menu.

The SMP Application Manager (SMPAM) main menu, Figure 9, is the next screen. Before instantly choosing option 10 and running SMP, the user needs to make sure the desired ship is the current one and the SMP input file has the correct loading, speeds, and appendages.

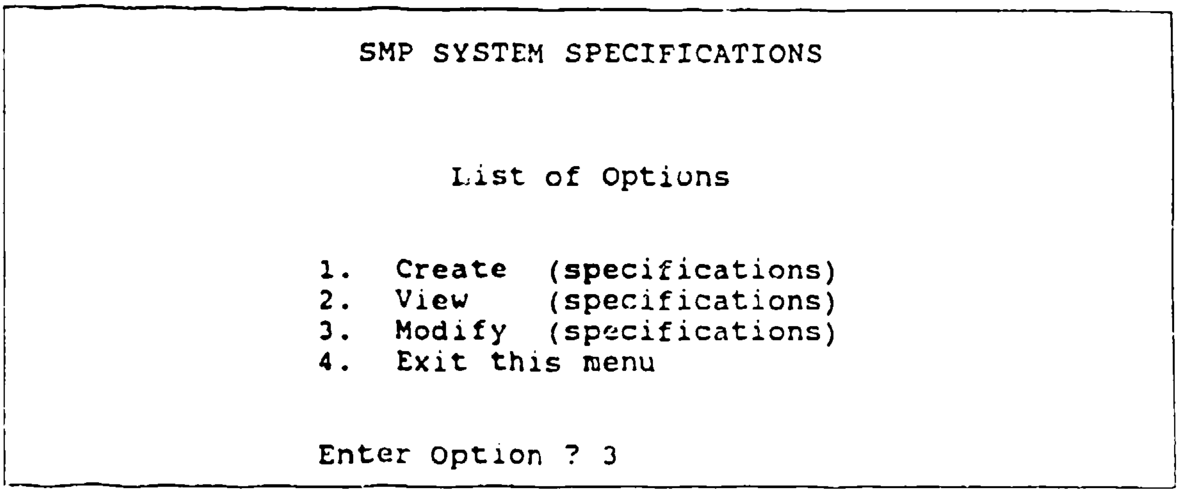
Setting current ship

If the input path were the default one, SMPINPUT, it would be possible to select option 4, Change ship and select the desired ship; however, this is not the case. The input file is in ALTINPUT\DESTROYR, so the input path is ALTINPUT. Instead of option 4, the user should choose option 2, SMP system specification, and then select option 3, Figure 10, to modify the system specifications. Details of SMP system specification are found in online help or in Appendix B.



```
                    SMP APPLICATIONS MANAGER

              Current Ship + Variant =  CVN68A


                        List of Options

 1. HELP                                  7. View ship hydrostatics
 2. SMP System Specification              8. Hull plot
 3. Change SMP data path                  9. View/Edit SMP input file
 4. Change ship                          10. Run  SMP
 5. View SMP Runlog file                 11. Polar plots
 6. View SMP Output file                 12. QUIT

    Enter Option ? 2
```

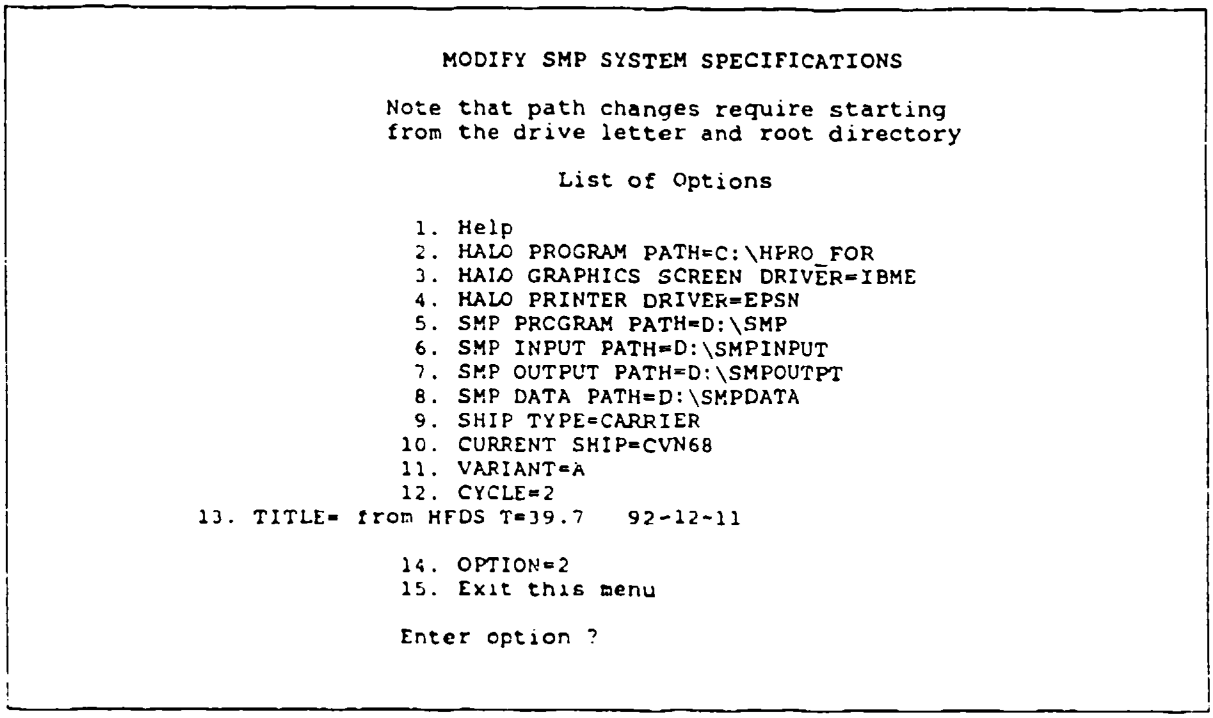
Fig. 9. SMP application manager main menu.



```
                  SMP SYSTEM SPECIFICATIONS


                       List of Options


              1.   Create  (specifications)
              2.   View    (specifications)
              3.   Modify  (specifications)
              4.   Exit this menu


              Enter Option ? 3
```

Fig. 10. SMP systems specifications menu.

The pertinent specifications to input path and current ship are: SMP input path, ship type, current ship, variant, and cycle. Here the user types the option number to change and enters the new values. The SMP input path is ALTINPUT; the ship type is the subdirectory DESTROYR; the current ship is DD965; the variant is H; and the cycle is 6. Figure 11 show the menu before any changes. Once all the changes are complete, Figure 12, the user enters 15 and then 4 to return to SMPAM main menu.



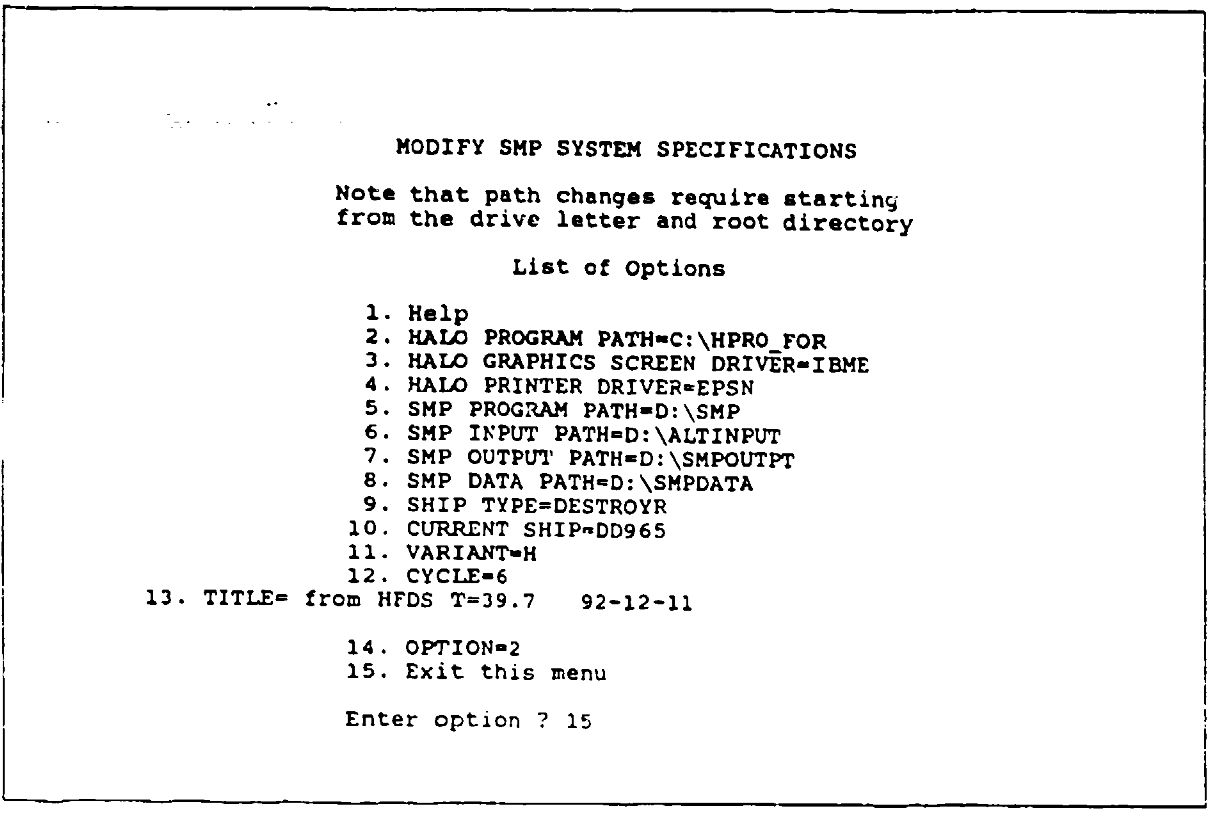
```
                          MODIFY SMP SYSTEM SPECIFICATIONS

                       Note that path changes require starting
                       from the drive letter and root directory

                                  List of Options

                         1. Help
                         2. HALO PROGRAM PATH=C:\HPRO_FOR
                         3. HALO GRAPHICS SCREEN DRIVER=IBME
                         4. HALO PRINTER DRIVER=EPSN
                         5. SMP PRCGRAM PATH=D:\SMP
                         6. SMP INPUT PATH=D:\SMPINPUT
                         7. SMP OUTPUT PATH=D:\SMPOUTPT
                         8. SMP DATA PATH=D:\SMPDATA
                         9. SHIP TYPE=CARRIER
                        10. CURRENT SHIP=CVN68
                        11. VARIANT=A
                        12. CYCLE=2
        13. TITLE= from HFDS T=39.7   92-12-11

                        14. OPTION=2
                        15. Exit this menu

                        Enter option ?
```

Fig. 11. Modify SMP systems specifications menu before changes.

Checking input

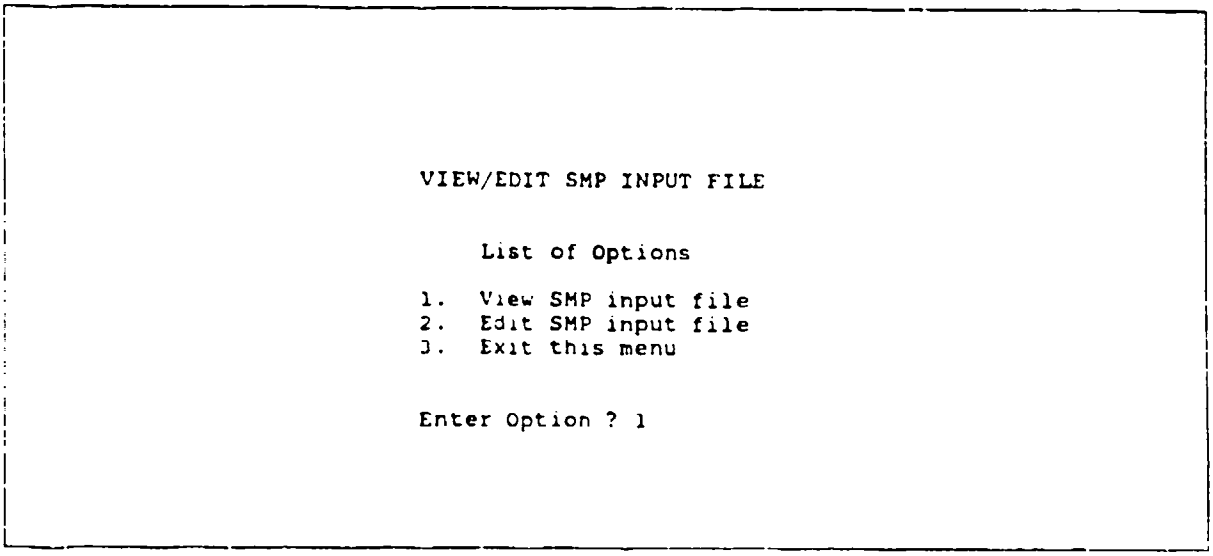
It is worthwhile to view the input before running SMP just to make sure the run is set up correctly. To check the input, select SMPAM main menu option 9, Figure 9, and option 1 of the next submenu, Figure 13.

Upon paging through the input file, we notice the speeds and sea states are wrong. The example problem has two speeds, 10 and 20 knots, and only one sea state. While *it is necessary to match speeds with the time history conditions*, the time history sea states are selected independently in STH. However, running four sea states is a waste



```
                   MODIFY SMP SYSTEM SPECIFICATIONS

                 Note that path changes require starting
                 from the drive letter and root directory

                            List of Options

                   1. Help
                   2. HALO PROGRAM PATH=C:\HPRO_FOR
                   3. HALO GRAPHICS SCREEN DRIVER=IBME
                   4. HALO PRINTER DRIVER=EPSN
                   5. SMP PROGRAM PATH=D:\SMP
                   6. SMP INPUT PATH=D:\ALTINPUT
                   7. SMP OUTPUT PATH=D:\SMPOUTPT
                   8. SMP DATA PATH=D:\SMPDATA
                   9. SHIP TYPE=DESTROYR
                  10. CURRENT SHIP=DD965
                  11. VARIANT=H
                  12. CYCLE=6
   13. TITLE= from HFDS T=39.7   92-12-11

                  14. OPTION=2
                  15. Exit this menu

                  Enter option ? 15
```

Fig. 12. Modify SMP systems specifications menu after changes.



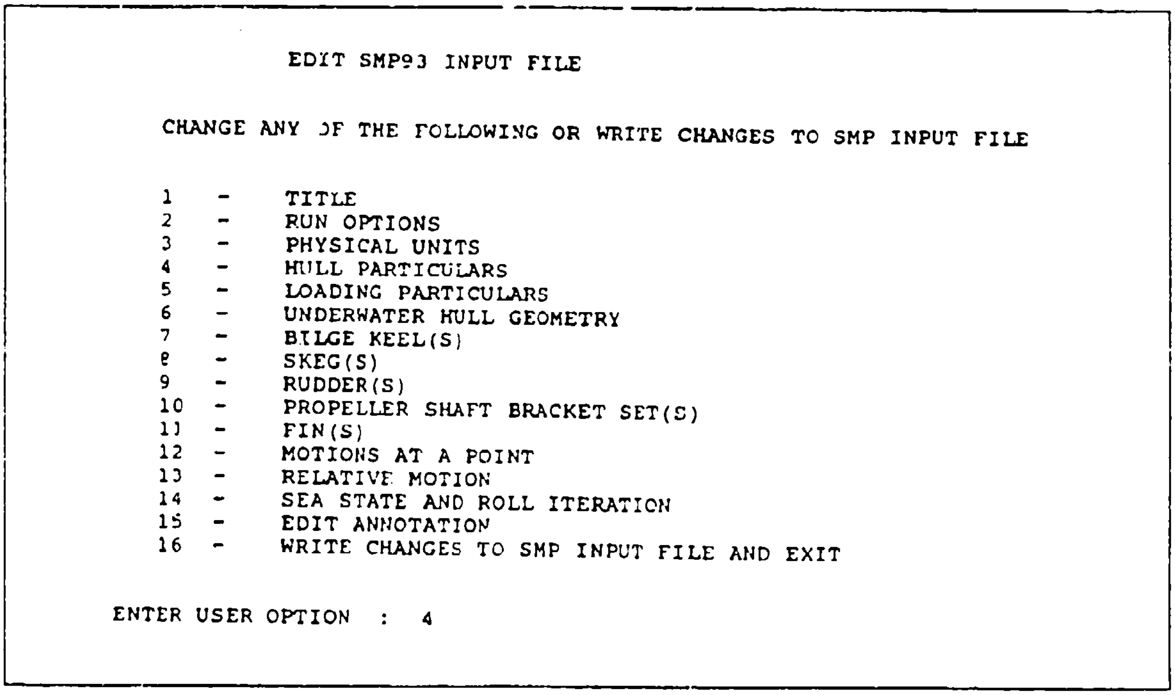
```
                    VIEW/EDIT SMP INPUT FILE


                         List of Options

                    1.  View SMP input file
                    2.  Edit SMP input file
                    3.  Exit this menu


                    Enter Option ? 1
```

Fig. 13. View/Edit SMP input file.

of time and disk space when only one is needed.

SMPEDIT

Returning to the previous menu, Figure 13, and selecting option 2 brings Figure 14 to the screen. SMPEDIT requires some knowledge of the SMP input format to know which variables are under what titles. Looking at Reference 1 reveals that ship speeds are under option 4, HULL PARTICULARS, and sea states are under option 14, SEA STATE AND ROLL ITERATION.

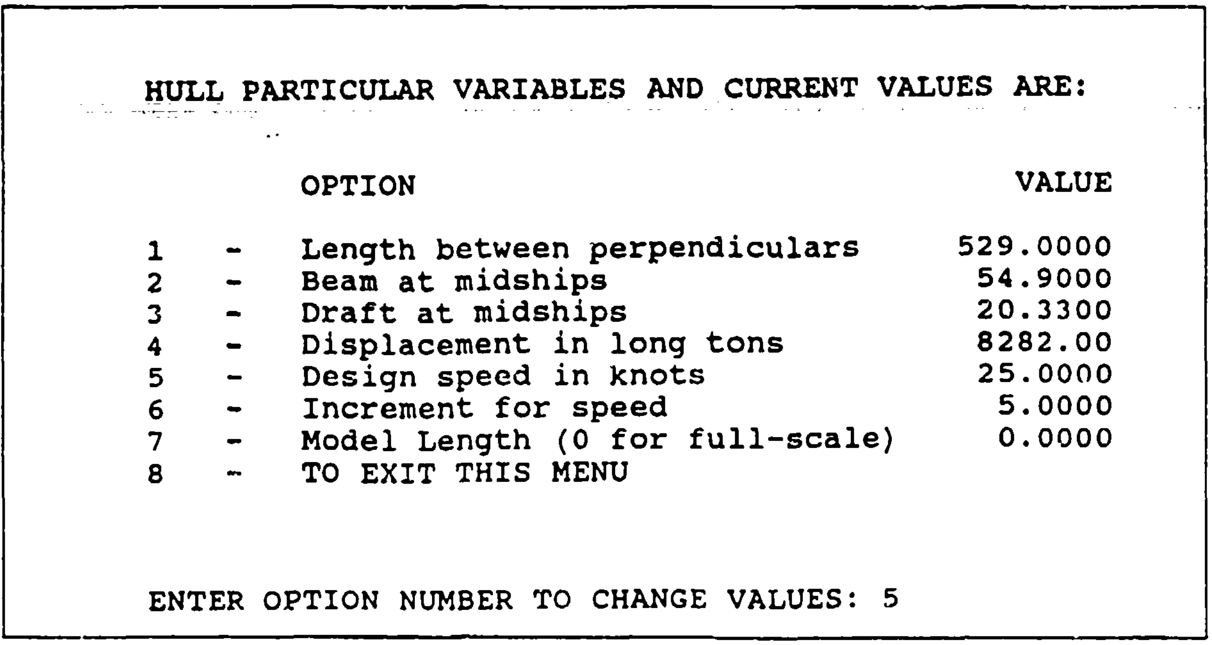


```
                    EDIT SMP93 INPUT FILE

      CHANGE ANY OF THE FOLLOWING OR WRITE CHANGES TO SMP INPUT FILE

      1    -    TITLE
      2    -    RUN OPTIONS
      3    -    PHYSICAL UNITS
      4    -    HULL PARTICULARS
      5    -    LOADING PARTICULARS
      6    -    UNDERWATER HULL GEOMETRY
      7    -    BILGE KEEL(S)
      8    -    SKEG(S)
      9    -    RUDDER(S)
      10   -    PROPELLER SHAFT BRACKET SET(S)
      11   -    FIN(S)
      12   -    MOTIONS AT A POINT
      13   -    RELATIVE MOTION
      14   -    SEA STATE AND ROLL ITERATION
      15   -    EDIT ANNOTATION
      16   -    WRITE CHANGES TO SMP INPUT FILE AND EXIT

    ENTER USER OPTION  :  4
```

Fig. 14. Edit SMP93-PC input file main menu.

Again the user selects the options to change and follows the submenu directions to make the desired changes. Selecting option 4, brings Figure 15 to the screen. Change the speeds to 20 knots maximum and 10-knot increments as stated in the problem statement.

Return to the SMPEDIT main menu and select option 14 to change the sea state. Figure 16 displays the other type of submenu. Here the user first determines a course of action, deleting, creating, or changing the data, and then specifies which variable to



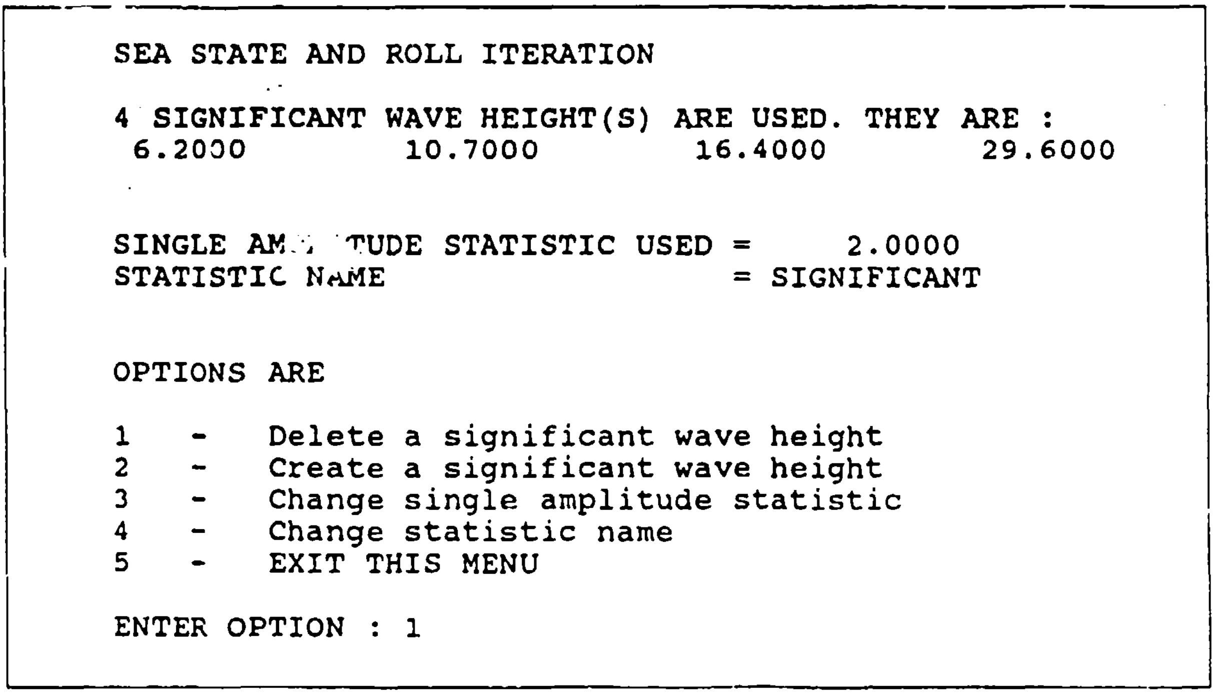
```
HULL PARTICULAR VARIABLES AND CURRENT VALUES ARE:

           OPTION                                VALUE

1    -     Length between perpendiculars      529.0000
2    -     Beam at midships                    54.9000
3    -     Draft at midships                   20.3300
4    -     Displacement in long tons          8282.00
5    -     Design speed in knots               25.0000
6    -     Increment for speed                  5.0000
7    -     Model Length (0 for full-scale)      0.0000
8    -     TO EXIT THIS MENU


ENTER OPTION NUMBER TO CHANGE VALUES: 5
```

Fig. 15. Hull particulars menu.

act on. After deleting the extra significant wave heights, 10.7, 16.4, and 29.6 feet, the user returns to the SMPEDIT main menu. In most cases, the same sea states are used for SMP and STH so most interesting conditions can be selected from the SMP output.

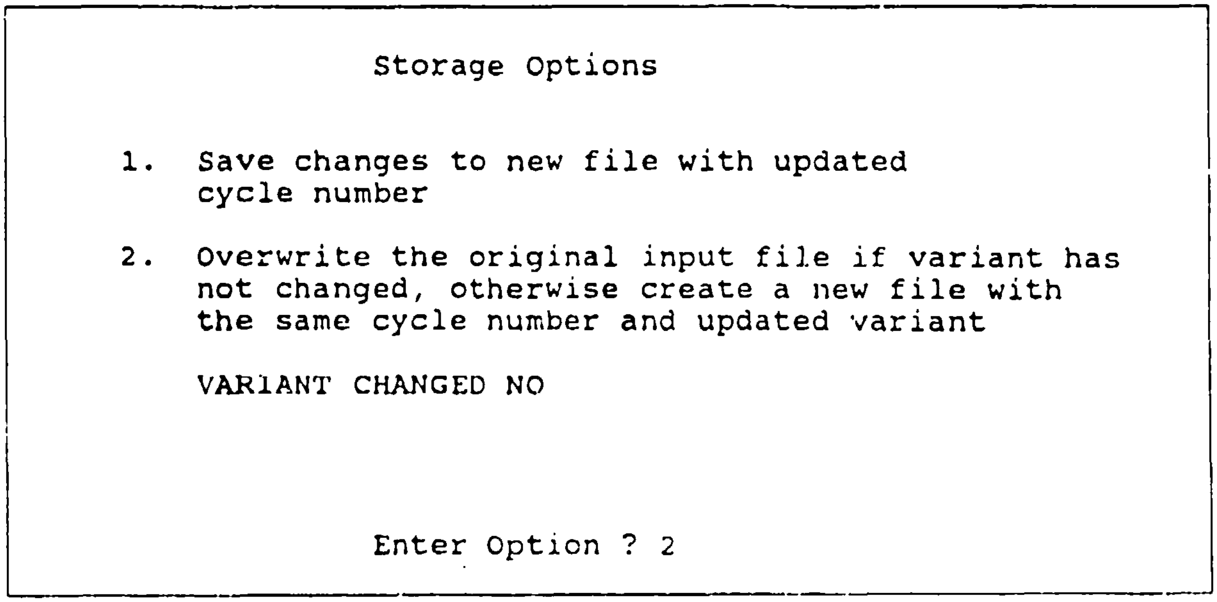
It is possible to delete in the SMP input file the sea state that would be later used when making time history runs. This is because STH uses the response transfer functions to calculate time histories, and the response transfer functions are virtually independent of sea state. The only non-linearity is in the lateral mode responses which depend on amplitude dependent roll damping. STH does type of roll iteration as SMP does to pick the correct roll transfer function. So it is possible to calculate the response transfer functions for one sea state and time histories for another sea state using those transfer functions.

Note that it is not necessary to specify the motion points in the SMP93-PC input file now to generate motion point time histories later. However, if there is some question as to the exact conditions to specify later, calculating motion point response with SMP would shed some light on which conditions are worst. To specify a point, select either option 12 or 13 and follow the submenu instructions. Further explanation is in Edit



```
SEA STATE AND ROLL ITERATION

4 SIGNIFICANT WAVE HEIGHT(S) ARE USED. THEY ARE :
 6.2000        10.7000        16.4000        29.6000


SINGLE AMPLITUDE STATISTIC USED =     2.0000
STATISTIC NAME                  = SIGNIFICANT


OPTIONS ARE

1    -    Delete a significant wave height
2    -    Create a significant wave height
3    -    Change single amplitude statistic
4    -    Change statistic name
5    -    EXIT THIS MENU

ENTER OPTION : 1
```

Fig. 16. Sea state and roll iteration menu.



```
                   Storage Options


     1.  Save changes to new file with updated
         cycle number

     2.  Overwrite the original input file if variant has
         not changed, otherwise create a new file with
         the same cycle number and updated variant

         VARIANT CHANGED NO



                   Enter Option ? 2
```

Fig. 17. SMPEDIT output storage options menu.

SMP input, page 12.

As a further input file check, it is useful to make a hydrostatic SMP run by entering 2 at the EDIT SMP input file main menu (Figure 14), for **Run Options** and setting OPTN equal to 1. Only after the hydrostatics run is it possible to **View ship hydrostatics** or **Hull plot** with the SMPAM. The rest of the run procedure is the same as the full SMP run (OPTN=2). After checking the hydrostatics, the user needs to edit the input file again to set OPTN to 2 or 3. For our example, we assume the input file is correct with respect to hydrostatics and hull shape.

Entering 16, for **Write changes to SMP input file and exit**, brings the user to Figure 17. The variant has not changed with just changing the speeds and sea states; select 2 to overwrite the file. The current ship is still DD965H6.INP. SMPEDIT updates the application manager's current ship automatically upon exiting SMPEDIT when the variant or cycle change.

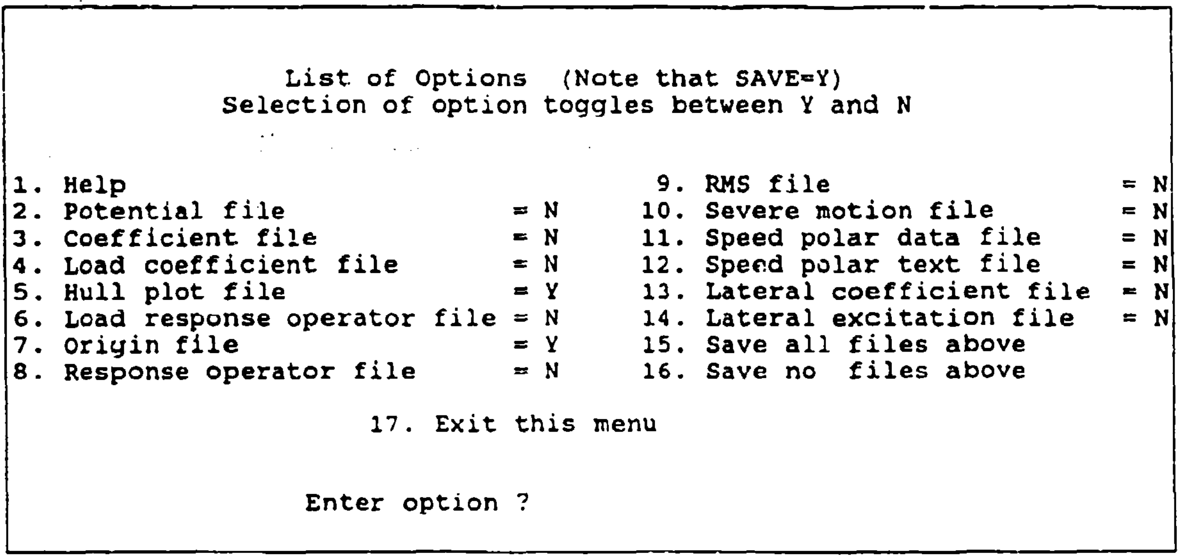
Running SMP93-PC

Now that the input file is correct, the user can run SMP93-PC, main menu option 10, and expect results. The screen displays a run log while SMP93-PC is running. This data is also written to the run log file, **SMPOUTPT\DESTROYR\DD965H6.TEX**. A run of one sea state and seven speeds run takes about 40 minutes on a 486DX/25+.

When SMP93-PC is through, Figure 18 appears on the screen giving the user the option to delete or save SMP93-PC output files. Selecting an option toggles between yes and no, or save and delete.

Save whatever files are necessary for future work, typically the hull plot, origin, and speed polar files. Generating time histories only requires the origin file, DD965H.ORG, which is automatically saved in subdirectory **SMPDATA\DD965**. Saving the hull plot file enables the user to use the SMPAM HULPLOT option. The speed polar files, SPD and SPT, are for generating speed polar plots.

Enter 17 to return to the SMPAM main menu (Figure 9) and then enter 12 to return to PREDICT main menu (Figure 8).



```
                    List of Options  (Note that SAVE=Y)
               Selection of option toggles between Y and N

1. Help                                   9. RMS file                  = N
2. Potential file                  = N   10. Severe motion file        = N
3. Coefficient file                = N   11. Speed polar data file     = N
4. Load coefficient file           = N   12. Speed polar text file     = N
5. Hull plot file                  = Y   13. Lateral coefficient file  = N
6. Load response operator file     = N   14. Lateral excitation file   = N
7. Origin file                     = Y   15. Save all files above
8. Response operator file          = N   16. Save no  files above

                          17. Exit this menu

                         Enter option ?
```

Fig. 18. Files to save after running SMP menu.

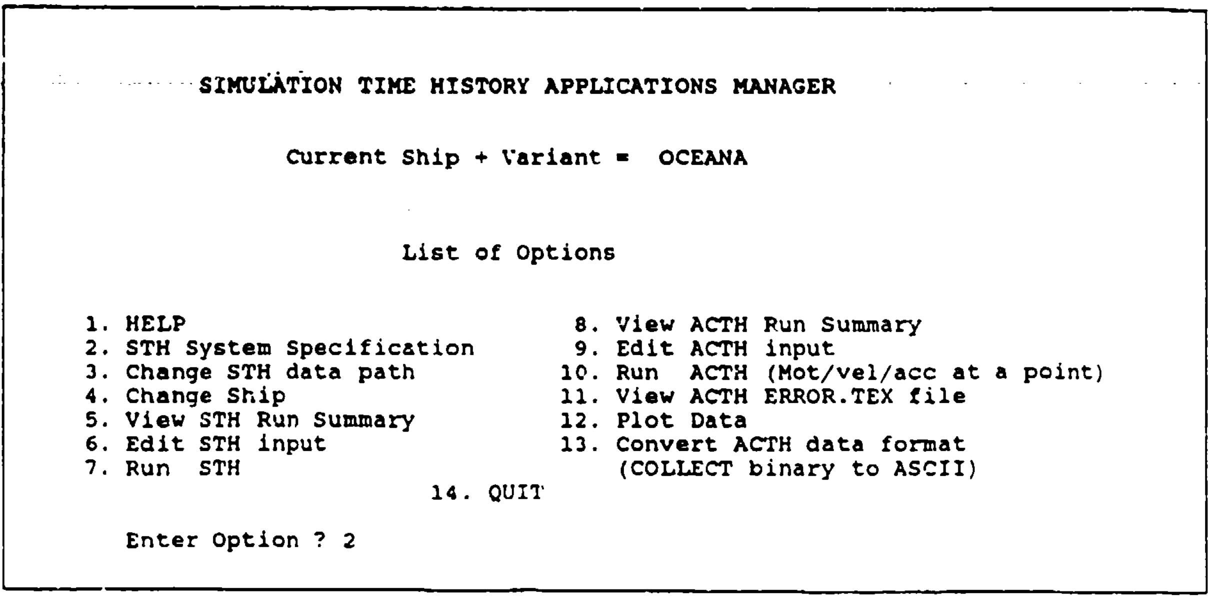
RUNNING STH

To launch the STH Application Manager (STHAM), enter 2 at PREDICT main menu (Figure 8). This brings the STHAM main menu, Figure 19, to the screen. Again it will be necessary to change the STH system specifications to match the SMP input path and current ship. This is carried out much like the SMPAM system specification changes and will not be demonstrated again.

STHEDIT

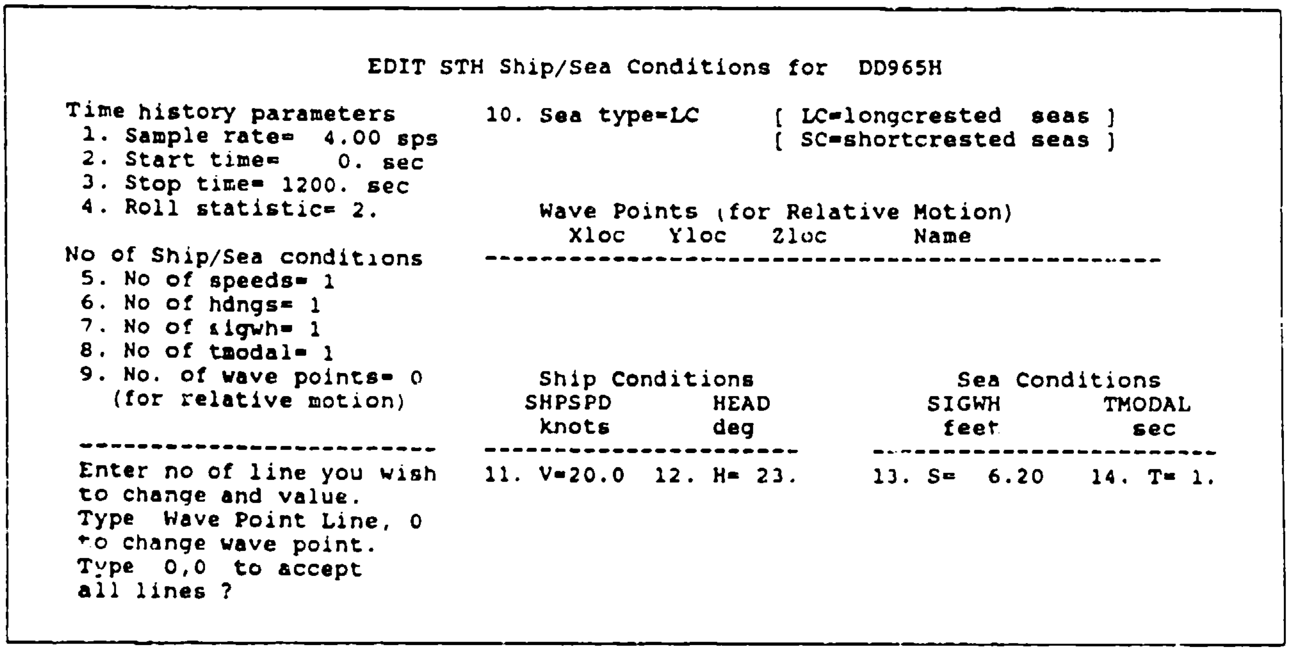
After correctly setting the current ship and paths, the next step is to generate a STH input file. Choose STHAM main menu option 6, **Edit STH input**, and bring up the EDIT STH Ship/Sea Conditions for menu, Figure 20. This screen has all pertinent data to make STH runs, but the default values may not be the desired ones. Each variable has a number to the left of it, which is referred to as a line number by this menu. As the instructions state, simply enter the number of the line to change and the new value, *separated by a comma*.

To get more choices for a wave point, ship, or sea condition line number, first increase the total number of choices for that variable. The line numbers automatically shift as the number of speeds, headings (hdngs), significant wave heights (sigwh), modal



```
                SIMULATION TIME HISTORY APPLICATIONS MANAGER

                      Current Ship + Variant =  OCEANA


                              List of Options

   1. HELP                               8. View ACTH Run Summary
   2. STH System Specification           9. Edit ACTH input
   3. Change STH data path              10. Run  ACTH (Mot/vel/acc at a point)
   4. Change Ship                       11. View ACTH ERROR.TEX file
   5. View STH Run Summary              12. Plot Data
   6. Edit STH input                    13. Convert ACTH data format
   7. Run  STH                              (COLLECT binary to ASCII)
                              14. QUIT

   Enter Option ? 2
```

Fig. 19. STH application manager main menu.



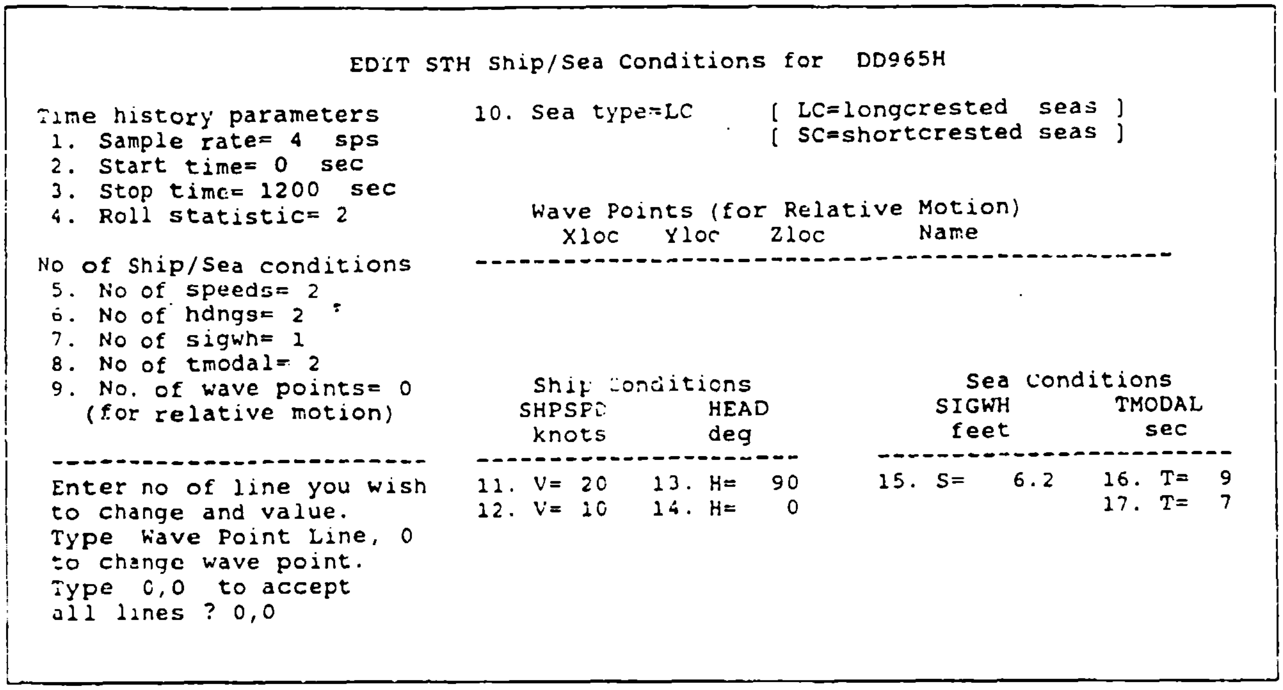
```
                         EDIT STH Ship/Sea Conditions for  DD965H

Time history parameters      10. Sea type=LC      [ LC=longcrested  seas ]
 1. Sample rate=  4.00 sps                        [ SC=shortcrested seas ]
 2. Start time=    0. sec
 3. Stop time= 1200. sec
 4. Roll statistic= 2.              Wave Points (for Relative Motion)
                                      Xloc   Yloc   Zloc      Name
No of Ship/Sea conditions    -----------------------------------------------
 5. No of speeds= 1
 6. No of hdngs= 1
 7. No of sigwh= 1
 8. No of tmodal= 1
 9. No. of wave points= 0         Ship Conditions              Sea Conditions
   (for relative motion)        SHPSPD        HEAD          SIGWH        TMODAL
                                 knots         deg           feet          sec
 ------------------------    ----------------------     -----------------------
 Enter no of line you wish   11. V=20.0  12. H= 23.     13. S=  6.20  14. T= 1.
 to change and value.
 Type  Wave Point Line, 0
 +o change wave point.
 Tvpe  0,0  to accept
 all lines ?
```

Fig. 20. Edit STII Ship/Sea conditions menu before changes.

periods (**tmodal**), and number of wave points change.

Relative motion time histories require some forethought because the wave points need to be specified in the STH input file and again in the ACTH input file. If the points specified in the two input files are not identical, ACTH will not run.

The two sea conditions for the example are: 20 knots, head seas (0 degrees), 6.2-foot significant wave height, and 9-second modal period; and 10 knots, beam seas (90 degrees), 6.2-foot significant wave height, and 7-second modal period. Again, note the significant wave height does not have to match that used for the SMP run.

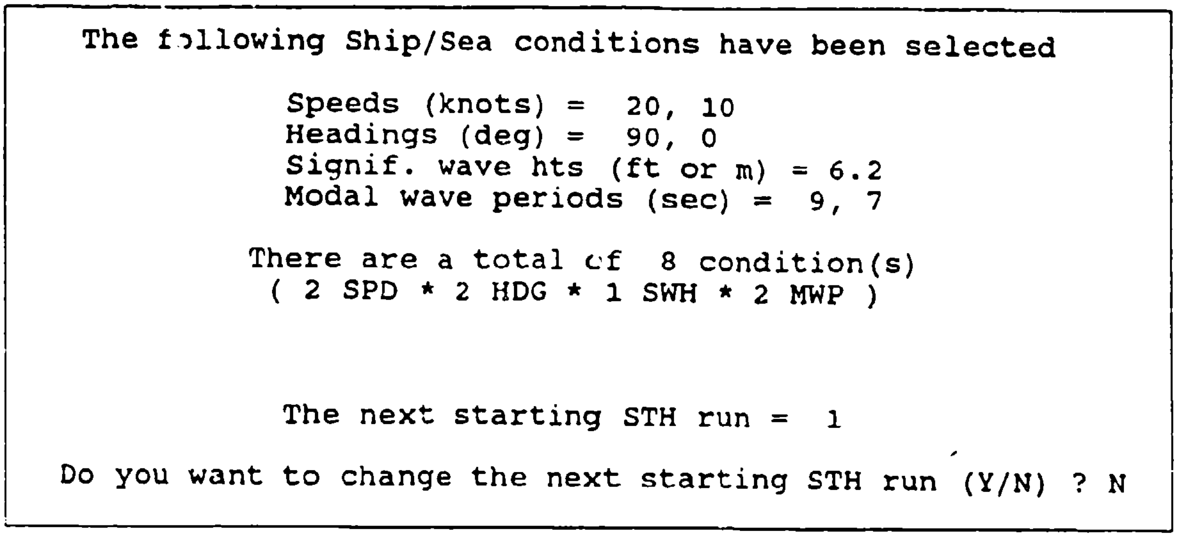


```
                      EDIT STH Ship/Sea Conditions for  DD965H

Time history parameters        10. Sea type=LC      [ LC=longcrested  seas ]
 1. Sample rate= 4  sps                             [ SC=shortcrested seas ]
 2. Start time= 0  sec
 3. Stop time= 1200  sec
 4. Roll statistic= 2              Wave Points (for Relative Motion)
                                     Xloc    Yloc    Zloc       Name
No of Ship/Sea conditions      ------------------------------------------------
 5. No of speeds= 2
 6. No of hdngs= 2
 7. No of sigwh= 1
 8. No of tmodal= 2
 9. No. of wave points= 0          Ship Conditions                Sea Conditions
   (for relative motion)          SHPSPD          HEAD          SIGWH          TMODAL
                                   knots           deg           feet            sec
 -------------------------     ----------------------     -------------------------
 Enter no of line you wish     11. V= 20    13. H=  90     15. S=    6.2    16. T=  9
 to change and value.          12. V= 10    14. H=   0                      17. T=  7
 Type  Wave Point Line, 0
 to change wave point.
 Type  0,0  to accept
 all lines ? 0,0
```

Fig. 21. Edit STH Ship/Sea conditions menu after changes.

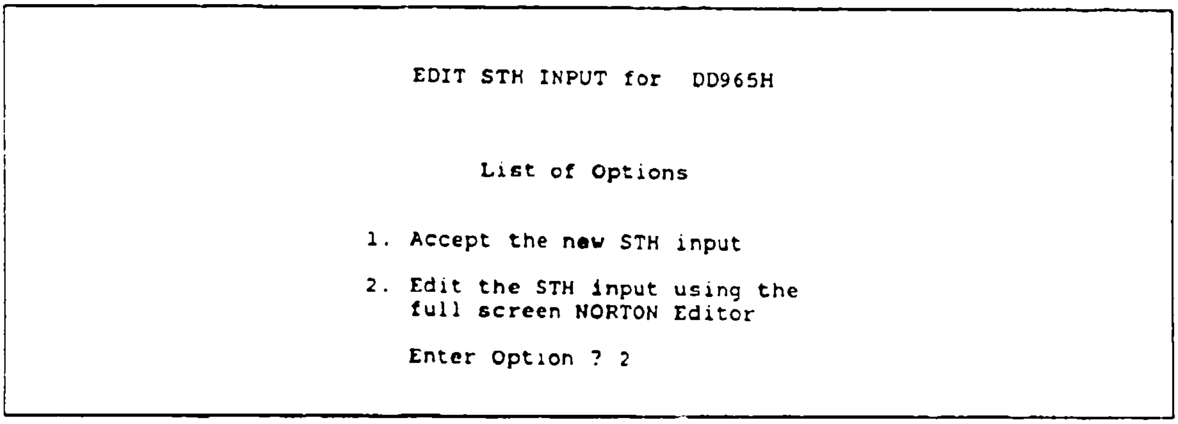
Enter **5,2** to set the number of speeds equal to two. Enter **12,10.0** so the two speeds are 10 and 20 knots. *It is important to include the decimal point when making changes.* Twenty knots already exists, being left over from a previous run. Increases in ship heading and modal period are made in the same fashion.

Once the changes are complete, Figure 21, entering 0,0 brings up Figure 22. PREDICT automatically generates sea conditions for the full matrix of possible combinations of ship speed, heading, significant wave height, and modal period and includes them in the input file. This screen displays the chosen sea conditions and number of possible combinations, in this case eight (2 speeds × 2 headings × 1 significant wave height × 2 modal periods). The new run numbers are consecutive to previous runs with this ship and variant, so existing data is not overwritten by accident. It is possible to override this by choosing another starting run number.



```
 The following Ship/Sea conditions have been selected

           Speeds (knots) =  20, 10
           Headings (deg) =  90, 0
           Signif. wave hts (ft or m) = 6.2
           Modal wave periods (sec) =  9, 7

         There are a total of  8 condition(s)
          ( 2 SPD * 2 HDG * 1 SWH * 2 MWP )



           The next starting STH run =  1

Do you want to change the next starting STH run (Y/N) ? N
```

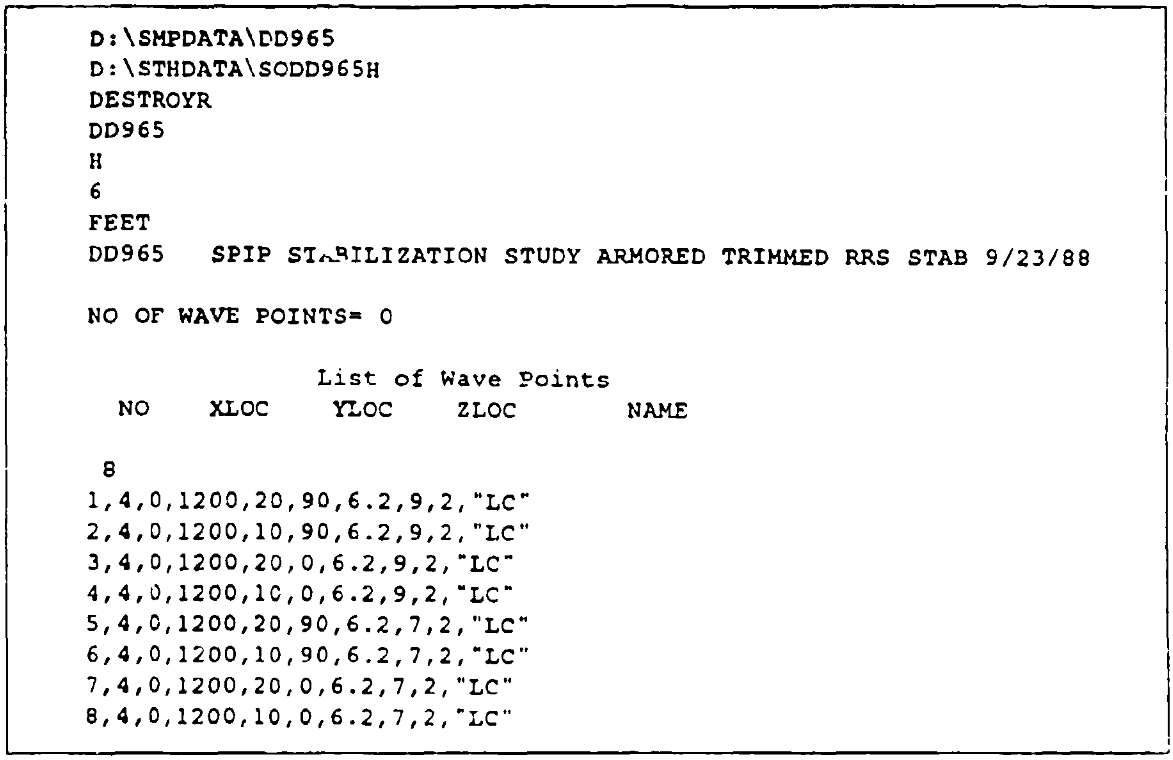
Fig. 22. STH ship/sea conditions and run number selection screen.



```
                EDIT STH INPUT for  DD965H


                    List of Options

             1. Accept the new STH input

             2. Edit the STH input using the
                full screen NORTON Editor

                Enter Option ? 2
```

Fig. 23. Edit STH input file menu.

The next screen, Figure 23, gives the user the option of one last editing of the input file using Norton^{TM} Editor[5]. Accepting the input file (option 2), means running the entire matrix of combinations. Usually, the conditions matrix has many superfluous entries. In our example, we have a matrix of eight conditions, but are only interested in two. In that case, the user should edit the file using Norton^{TM} Editor (option 2). In Norton^{TM} Editor, simply delete the unwanted sea conditions, change the total number of sea conditions to the correct value, and renumber the remaining sea conditions. While this may seem tedious, it actually takes less time, uses less disk space, and is much easier than keeping track of all the extra runs.

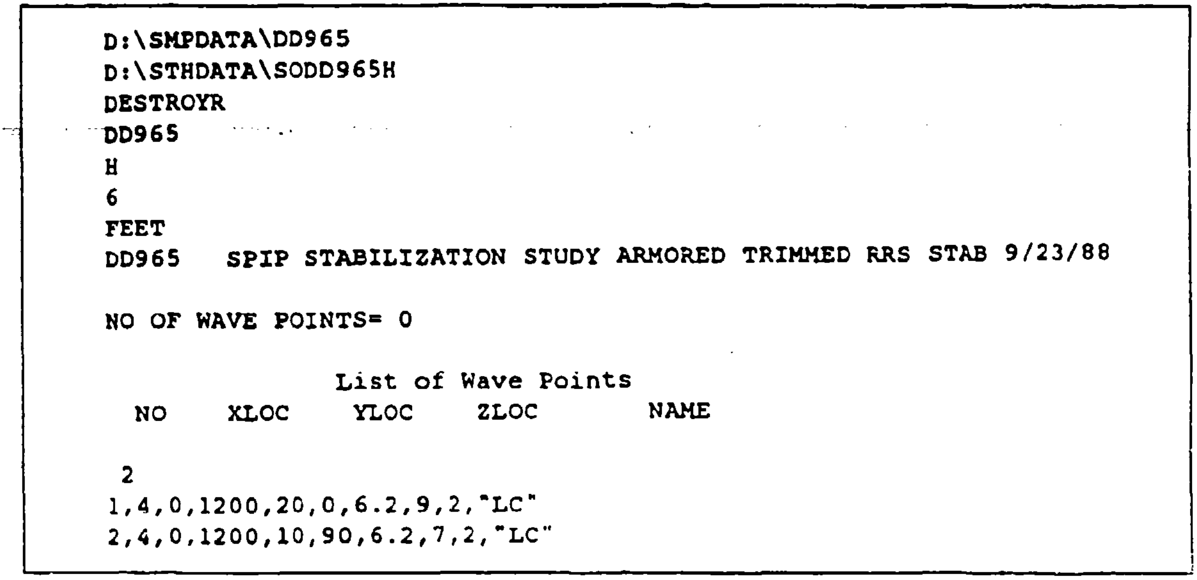


```
D:\SMPDATA\DD965
D:\STHDATA\SODD965H
DESTROYR
DD965
H
6
FEET
DD965    SPIP STABILIZATION STUDY ARMORED TRIMMED RRS STAB 9/23/88

NO OF WAVE POINTS= 0

               List of Wave Points
  NO     XLOC     YLOC     ZLOC         NAME

 8
1,4,0,1200,20,90,6.2,9,2,"LC"
2,4,0,1200,10,90,6.2,9,2,"LC"
3,4,0,1200,20,0,6.2,9,2,"LC"
4,4,0,1200,10,0,6.2,9,2,"LC"
5,4,0,1200,20,90,6.2,7,2,"LC"
6,4,0,1200,10,90,6.2,7,2,"LC"
7,4,0,1200,20,0,6.2,7,2,"LC"
8,4,0,1200,10,0,6.2,7,2,"LC"
```

Fig. 24. STH input file before deleting superfluous conditions.

The example has six unwanted conditions, so the user should choose to edit the file (option 2). Figures 24 and 25 show the input file, STH.INP, before and after the changes. PREDICT also updates the ship specific and generic STHAM input file, DD965H.INP and STHAM.INP, at this time.

If editing STH.INP outside the application manager, change the number of STH runs in STHAM.INP and STH*Root Variant*.INP to match that in STH.INP.



```
D:\SMPDATA\DD965
D:\STHDATA\SODD965H
DESTROYR
DD965
H
6
FEET
DD965   SPIP STABILIZATION STUDY ARMORED TRIMMED RRS STAB 9/23/88

NO OF WAVE POINTS= 0

                 List of Wave Points
  NO    XLOC     YLOC     ZLOC         NAME

 2
1,4,0,1200,20,0,6.2,9,2,"LC"
2,4,0,1200,10,90,6.2,7,2,"LC"
```

Fig. 25. STH input file after deleting superfluous conditions.

Running STH

STH generates longcrested or shortcrested six-degree-of-freedom time histories at the origin (LCG at waterline on centerline). Longcrested waves have all the wave energy coming from one direction. STH shortcrested waves use a cosine squared spreading function to spread the wave energy over ±90° from the primary heading. Longcrested runs are very quick, on the order of a minute per condition, and shortcrested runs take about 10 minutes per condition on a 486DX/25+.

To run STH with the current ship and input file, the user selects STHAM main menu option 7, **Run STH**. STH writes progress reports to the screen and to the STH run summary file, **STHLOG.TEX**, while it runs. The time histories are saved to the files **STHDATA\SODD965H\SR1.DAT** and **SR2.DAT**. When through, STH returns to the STHAM main menu.

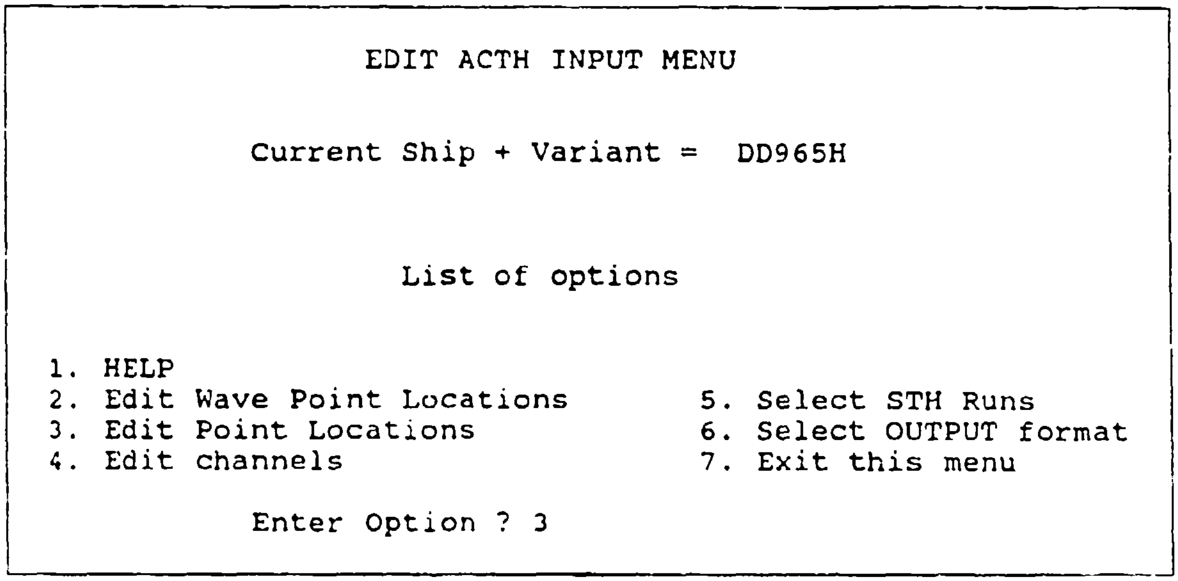
RUNNING ACTH

After STH calculates the origin time histories, the next steps in generating motion at a point time histories are to generate an ACTH input file and run ACTH.

ACTHEDIT

From the STHAM main menu, select option 9, **Edit ACTH input**, to bring up the first ACTHEDIT submenu, Figure 26. The ACTH editor works much the same way as SMPEDIT. There is the first submenu with the listing of the main parts of the input file, e.g. point locations and channels. The user selects the option and follows the instructions for further defining the variables, e.g. entering point locations. The point location for the example is a boom tip off the destroyer fan tail, station 20.1, 15 feet starboard of centerline, and 40 feet up from the baseline.

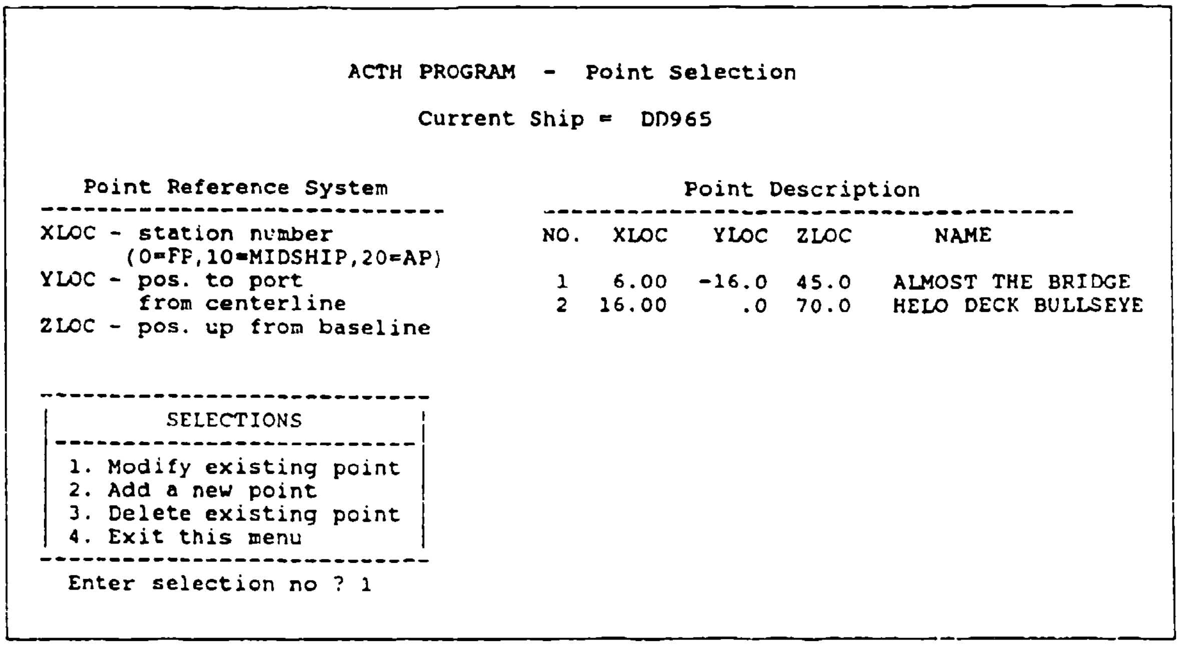
ACTH uses the same point location coordinate system as SMP[1], with the origin at the forward perpendicular, on the centerline and baseline. The x direction is longitudinal along the hull and uses station numbering, with the forward perpendicular being station 0 and station numbers increasing aft. Y is positive to port and z is positive up.



```
                    EDIT ACTH INPUT MENU

              Current Ship + Variant =  DD965H


                       List of options

 1. HELP
 2. Edit Wave Point Locations        5. Select STH Runs
 3. Edit Point Locations             6. Select OUTPUT format
 4. Edit channels                    7. Exit this menu

              Enter Option ? 3
```

Fig. 26. Edit ACTH input menu.

Selecting option 3 from the **EDIT ACTH INPUT MENU** brings up the point selection menu, Figure 27, which displays points from the previous run. Select option 1 to modify an existing point and bring up the **Modify existing point** submenu, Figure 28. Enter the number of the point to modify. Then at the new prompt, Figure 29, enter the new location and name, separated by commas. PREDICT returns to the previous menu so the user may enter the next point to modify. Enter 0 to return to Point Selection menu. Now the user has the choice to delete the extra locations or to simply not save those channels.when selecting the channels. Delete and add points in the same manner as modifying them. Entering a 4 returns the user to the main ACTH edit menu, Figure 26.

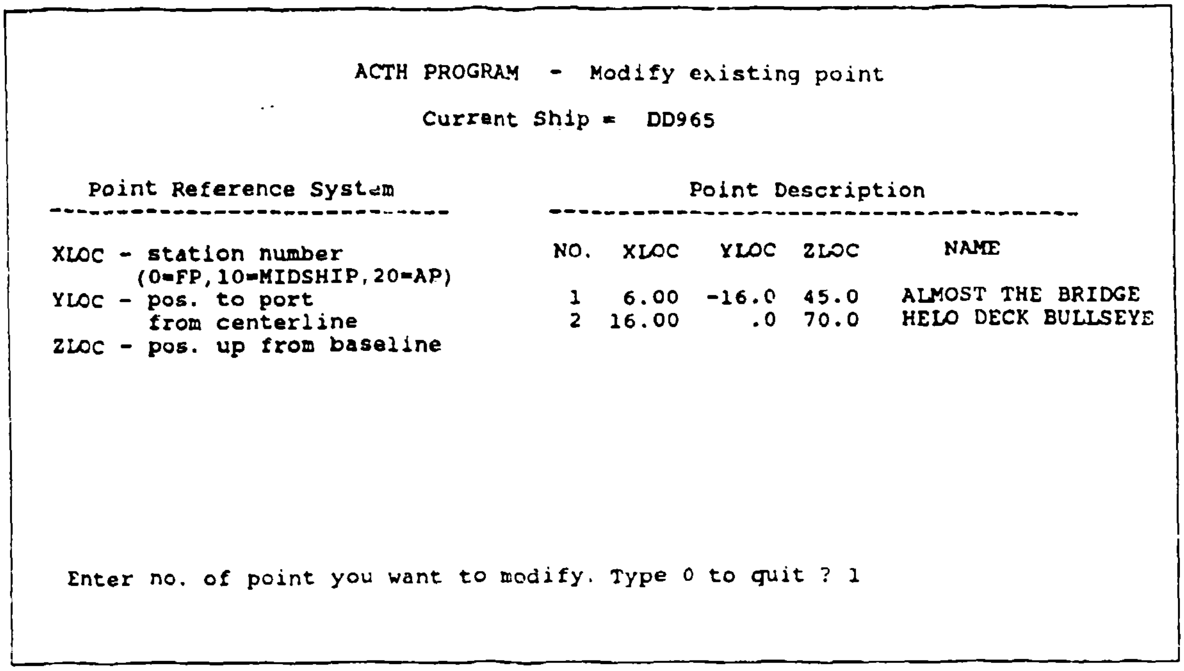


```
                        ACTH PROGRAM  -  Point Selection

                           Current Ship =  DD965

   Point Reference System                         Point Description
 ----------------------------         ------------------------------------------
 XLOC - station number                 NO.   XLOC    YLOC   ZLOC       NAME
       (0=FP,10=MIDSHIP,20=AP)
 YLOC - pos. to port                    1    6.00   -16.0   45.0   ALMOST THE BRIDGE
        from centerline                 2   16.00      .0   70.0   HELO DECK BULLSEYE
 ZLOC - pos. up from baseline


 ---------------------------
 |        SELECTIONS        |
 |--------------------------|
 | 1. Modify existing point |
 | 2. Add a new point       |
 | 3. Delete existing point |
 | 4. Exit this menu        |
 ---------------------------
   Enter selection no ? 1
```

Fig. 27. ACTH program point selection menu.

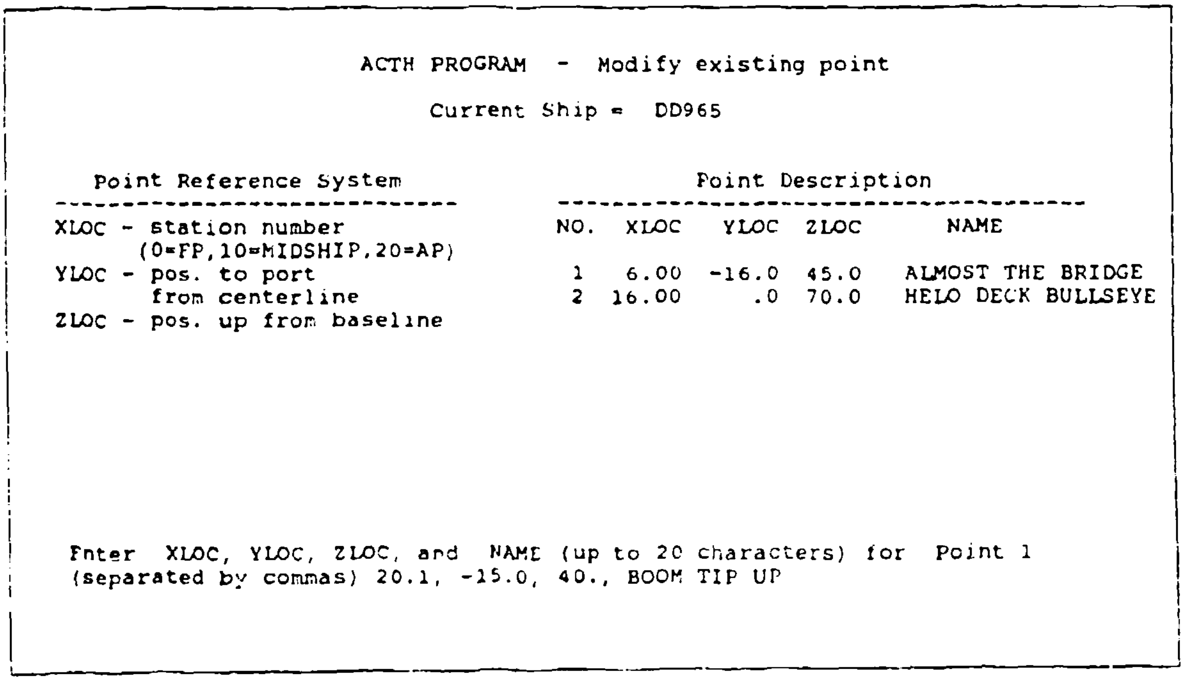
Edit wave point locations in the same fashion as ship point locations. If relative motion (wave) points were specified in the STH runs, it is possible to calculate relative motion time histories with ACTH as long as the point locations are the same. But just because the STH runs have wave poi its. does not mean the ACTH runs have to have them also. *However, in cases where the previous ACTH run asked for wave points, and the current STH run does not, then the previous wave point locations need to be deleted.*

The user chooses which response time histories to save by editing the channels, EDIT ACTH main menu option 4. Figure 30 shows the Channel selection menu. Again channel selections from the previous run are left over and might need modification. Each of the menu selections has its own submenus which prompt the user for a channel



```
                    ACTH PROGRAM  -  Modify existing point

                           Current Ship =  DD965

   Point Reference System                    Point Description
 ---------------------------         ------------------------------------
 XLOC - station number               NO.  XLOC    YLOC   ZLOC       NAME
      (0=FP,10=MIDSHIP,20=AP)
 YLOC - pos. to port                  1    6.00  -16.0   45.0   ALMOST THE BRIDGE
        from centerline               2   16.00     .0   70.0   HELO DECK BULLSEYE
 ZLOC - pos. up from baseline




  Enter no. of point you want to modify. Type 0 to quit ? 1
```

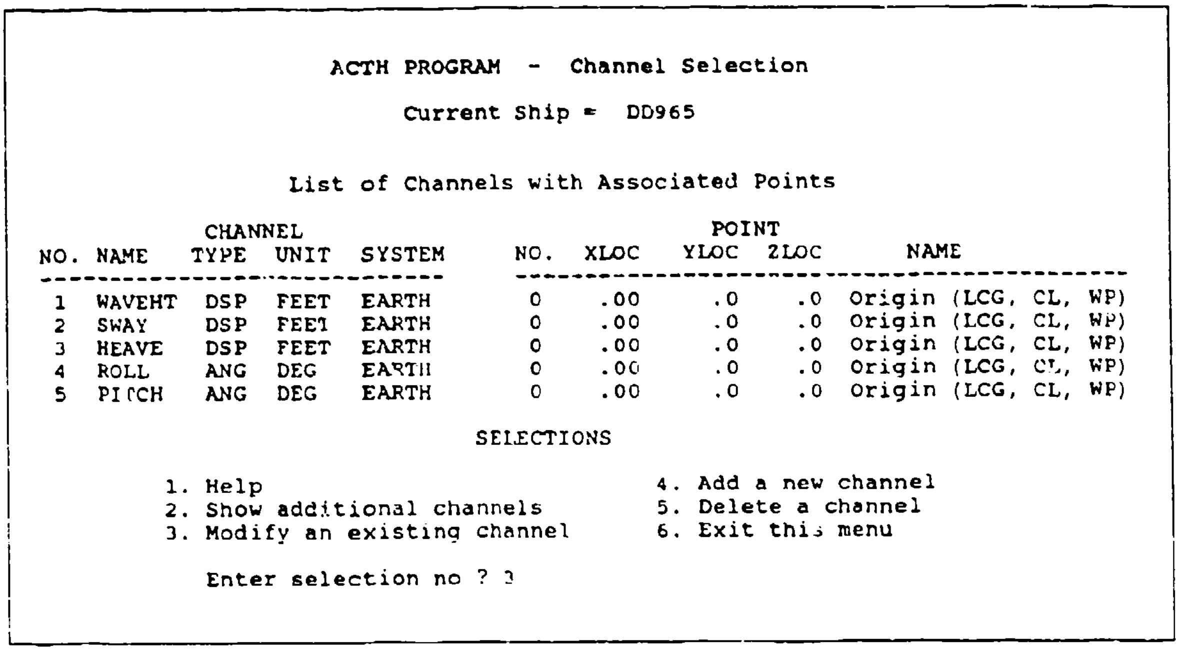
Fig. 28. ACTH program point selection - modify existing point submenu.



```
                    ACTH PROGRAM  -  Modify existing point

                           Current Ship =  DD965

   Point Reference System                    Point Description
 ---------------------------         ------------------------------------
 XLOC - station number               NO.  XLOC    YLOC   ZLOC       NAME
      (0=FP,10=MIDSHIP,20=AP)
 YLOC - pos. to port                  1    6.00  -16.0   45.0   ALMOST THE BRIDGE
        from centerline               2   16.00     .0   70.0   HELO DECK BULLSEYE
 ZLOC - pos. up from baseline




  Enter  XLOC, YLOC, ZLOC, and  NAME (up to 20 characters) for  Point 1
  (separated by commas) 20.1, -15.0, 40., BOOM TIP UP
```

Fig. 29. ACTH program point selection - entering new point submenu.

number to operate on and which response to store in that channel, i.e. whether it is vertical, lateral, or longitudinal; displacement, velocity, acceleration, or force. See **Channel selection**, page 23, for the possible channel choices and brief descriptions. Figures 31-35 show the menus used when modifying the channel selection to include vertical acceleration of the boom tip.



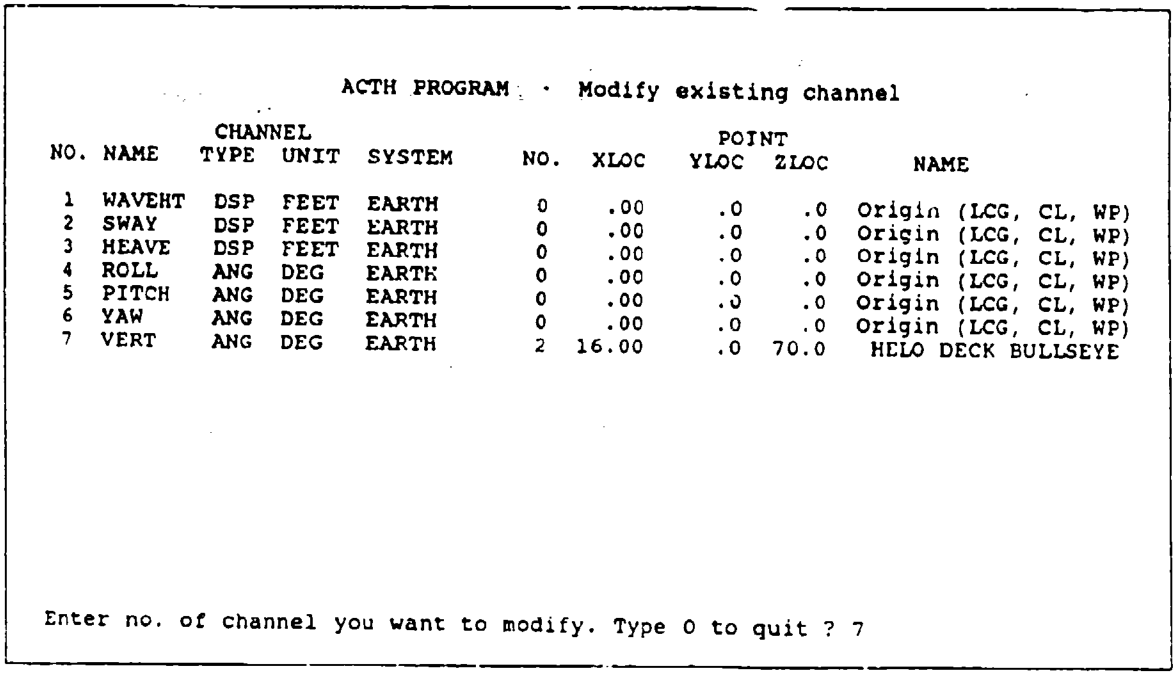
```
                     ACTH PROGRAM  -  Channel Selection
                          Current Ship =  DD965

                   List of Channels with Associated Points

              CHANNEL                                POINT
NO. NAME    TYPE  UNIT   SYSTEM       NO.  XLOC    YLOC  ZLOC        NAME
------------------------------       ----------------------------------------------
 1  WAVEHT  DSP   FEET   EARTH         0    .00     .0     .0   Origin (LCG, CL, WP)
 2  SWAY    DSP   FEET   EARTH         0    .00     .0     .0   Origin (LCG, CL, WP)
 3  HEAVE   DSP   FEET   EARTH         0    .00     .0     .0   Origin (LCG, CL, WP)
 4  ROLL    ANG   DEG    EARTH         0    .00     .0     .0   Origin (LCG, CL, WP)
 5  PITCH   ANG   DEG    EARTH         0    .00     .0     .0   Origin (LCG, CL, WP)

                                 SELECTIONS

           1. Help                              4. Add a new channel
           2. Show additional channels          5. Delete a channel
           3. Modify an existing channel        6. Exit this menu

              Enter selection no ? 3
```

Fig. 30. ACTH program channel selection menu.

EDIT ACTH input menu (Figure 26) option 5, `Select STH Runs`, selects the STH runs to use when calculating the ACTH time histories. Figure 36 shows the `Select STH runs` menu. This is the same method of selecting sea conditions used with the STH editor. PREDICT gets the ship/sea conditions from current ship subdirectory. The desired ship/sea conditions are flagged (*) and the undesired ones unflagged. ACTH runs will be made for whichever STH runs match the flagged sea conditions. *When flagging or unflagging, enter the number exactly as shown on the screen, including decimal point and separate with a comma.* For example, to flag, enter `30.0,F`.

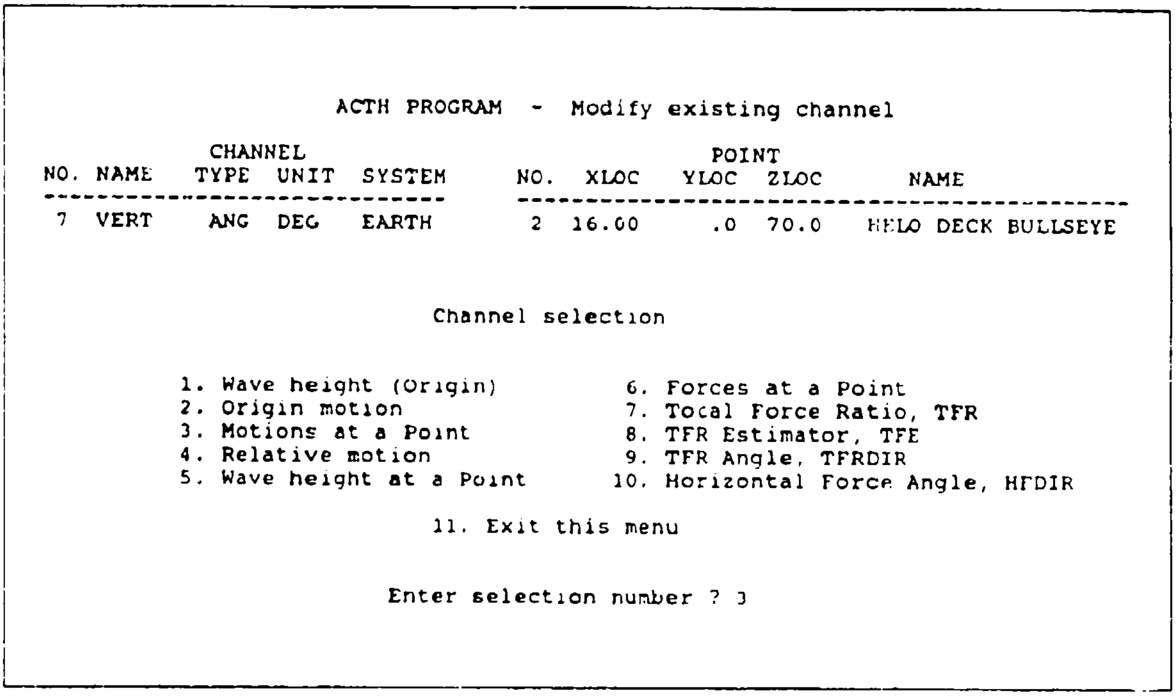
The output can be either binary or ASCII in Option 6 of the EDIT ACTH main menu (Figure 26). The default output format is binary.

EDIT ACTH main menu (Figure 26) Option 7, `Exit this menu`, brings up the



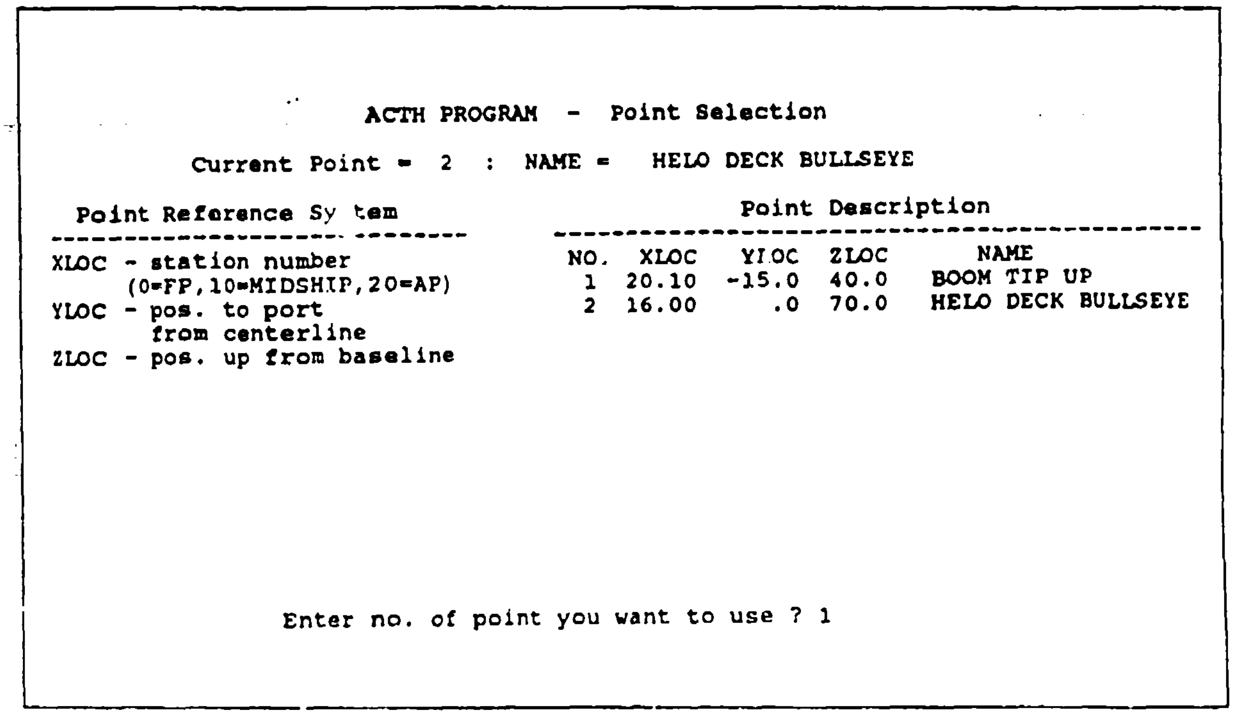
```
                    ACTH PROGRAM  -  Modify existing channel

              CHANNEL                                 POINT
NO. NAME    TYPE  UNIT   SYSTEM      NO.   XLOC    YLOC   ZLOC          NAME

 1  WAVEHT   DSP  FEET   EARTH        0     .00      .0     .0    Origin (LCG, CL, WP)
 2  SWAY     DSP  FEET   EARTH        0     .00      .0     .0    Origin (LCG, CL, WP)
 3  HEAVE    DSP  FEET   EARTH        0     .00      .0     .0    Origin (LCG, CL, WP)
 4  ROLL     ANG  DEG    EARTH        0     .00      .0     .0    Origin (LCG, CL, WP)
 5  PITCH    ANG  DEG    EARTH        0     .00      .0     .0    Origin (LCG, CL, WP)
 6  YAW      ANG  DEG    EARTH        0     .00      .0     .0    Origin (LCG, CL, WP)
 7  VERT     ANG  DEG    EARTH        2   16.00      .0   70.0     HELO DECK BULLSEYE







Enter no. of channel you want to modify. Type 0 to quit ? 7
```

Fig. 31. ACTH program modify existing channel menu.



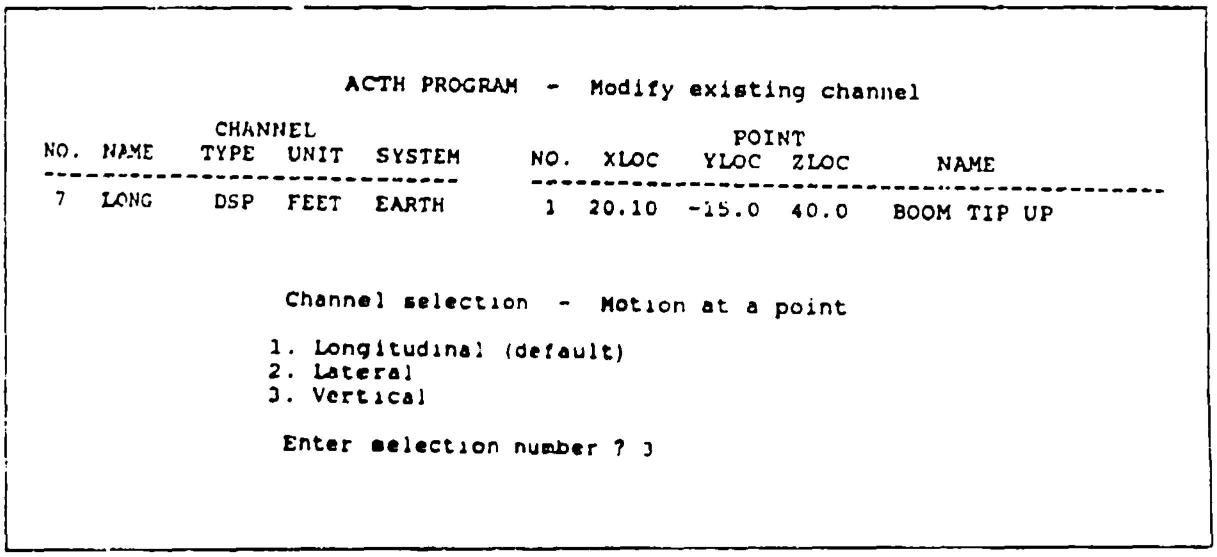
```
                    ACTH PROGRAM  -  Modify existing channel

              CHANNEL                                 POINT
NO. NAME    TYPE  UNIT   SYSTEM      NO.   XLOC    YLOC   ZLOC          NAME
-----------------------------------  -------------------------------------------------
 7  VERT     ANG  DEG    EARTH        2   16.00      .0   70.0     HELO DECK BULLSEYE


                              Channel selection

              1. Wave height (Origin)         6. Forces at a Point
              2. Origin motion                7. Tocal Force Ratio, TFR
              3. Motions at a Point           8. TFR Estimator, TFE
              4. Relative motion              9. TFR Angle, TFRDIR
              5. Wave height at a Point      10. Horizontal Force Angle, HFDIR

                          11. Exit this menu


                        Enter selection number ? 3
```

Fig. 32. Modify existing channel submenu - ACTH channel selection menu.



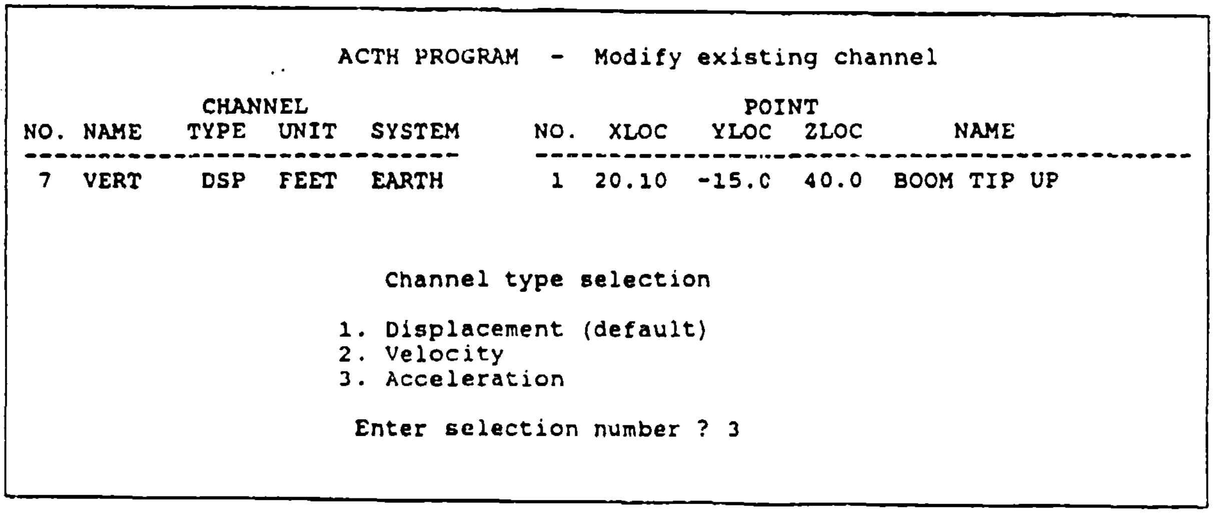
```
                    ACTH PROGRAM  -  Point Selection

           Current Point =  2   :  NAME =   HELO DECK BULLSEYE

 Point Reference Sy tem                         Point Description
 ----------------------- --------      ------------------------------------------
XLOC - station number                   NO.  XLOC    YIOC   ZLOC       NAME
     (0=FP,10=MIDSHIP,20=AP)             1   20.10   -15.0  40.0   BOOM TIP UP
YLOC - pos. to port                      2   16.00      .0  70.0   HELO DECK BULLSEYE
      from centerline
ZLOC - pos. up from baseline




             Enter no. of point you want to use ? 1
```

Fig. 33. Modify existing channel submenu - ACTH program point selection menu.



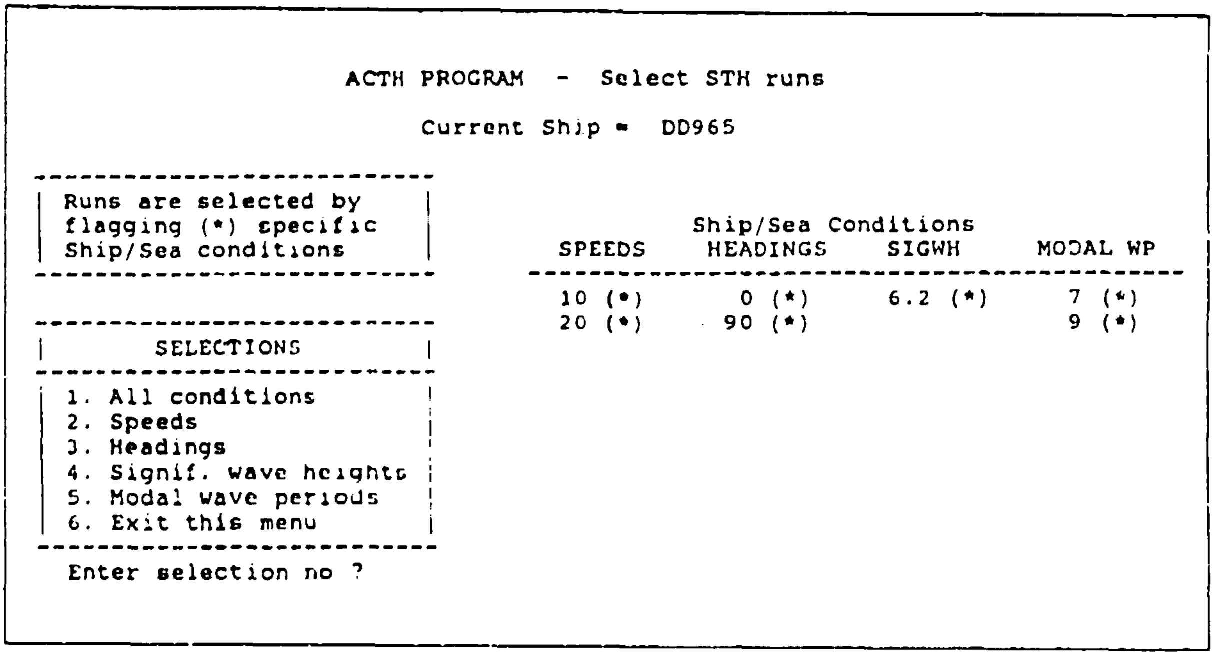
```
                     ACTH PROGRAM  -  Modify existing channel

             CHANNEL                                     POINT
NO. NAME   TYPE  UNIT  SYSTEM       NO.  XLOC   YLOC  ZLOC        NAME
------------------------------      ------------------------------------------
 7  LONG    DSP  FEET  EARTH         1  20.10  -15.0  40.0    BOOM TIP UP


                 Channel selection  -  Motion at a point

                1. Longitudinal (default)
                2. Lateral
                3. Vertical

                 Enter selection number ? 3
```

Fig. 34. Modify existing channel submenu - ACTH program motion at a point mode menu.



```
                         ACTH PROGRAM  -  Modify existing channel

           CHANNEL                                       POINT
NO. NAME   TYPE  UNIT   SYSTEM      NO.  XLOC    YLOC   ZLOC         NAME
--------------------------------    -------------------------------------------------
 7  VERT    DSP  FEET   EARTH        1  20.10  -15.0   40.0  BOOM TIP UP


                            Channel type selection

                          1. Displacement (default)
                          2. Velocity
                          3. Acceleration

                           Enter selection number ? 3
```

Fig. 35. Modify existing channel submenu - ACTH program response selection menu.



```
                        ACTH PROGRAM  -  Select STH runs

                            Current Ship =  DD965

 --------------------------
| Runs are selected by     |
| flagging (*) specific    |              Ship/Sea Conditions
| Ship/Sea conditions      |      SPEEDS    HEADINGS     SIGWH       MODAL WP
 --------------------------      -------------------------------------------
                                  10 (*)       0 (*)     6.2 (*)       7 (*)
 --------------------------       20 (*)      90 (*)                   9 (*)
|        SELECTIONS        |
 --------------------------
| 1. All conditions        |
| 2. Speeds                |
| 3. Headings              |
| 4. Signif. wave heights  |
| 5. Modal wave periods    |
| 6. Exit this menu        |
 --------------------------
  Enter selection no ?
```

Fig. 36. ACTH program STH run selection menu.

same series of questions as exiting STHEDIT (Figure 23), i.e. whether to number the runs consecutively and whether to accept the input file. The input file can be edited using Norton^{TM} Editor from within PREDICT, or outside of PREDICT altogether. The files STH\ACTH.INP, STHAM.INP, and DD965H.INP are updated at this point and the user returns to the STHAM main menu, Figure 19.

Running ACTH

Select STHAM main menu (Figure 19) option 10, Run ACTH (Mot/vel/acc at a point), to run ACTH with the current ship and input file. ACTH writes progress reports to the screen and DR*.TEX, and to the ACTH run summary file, TRIALLOG.TEX, while it runs. The output files DR1.* and DR2.*[†] are written to the COLLECT data directory and SP*Root Variant* subdirectory, DATA\SPDD965H.

ACTH runs very quickly, on the order of 1 minute per condition and, when through, returns to the STHAM main menu. If an ACTH run fails to finish, viewing ERROR.TEX, STHAM main menu (Figure 19) option 11, can provide clues as to why.

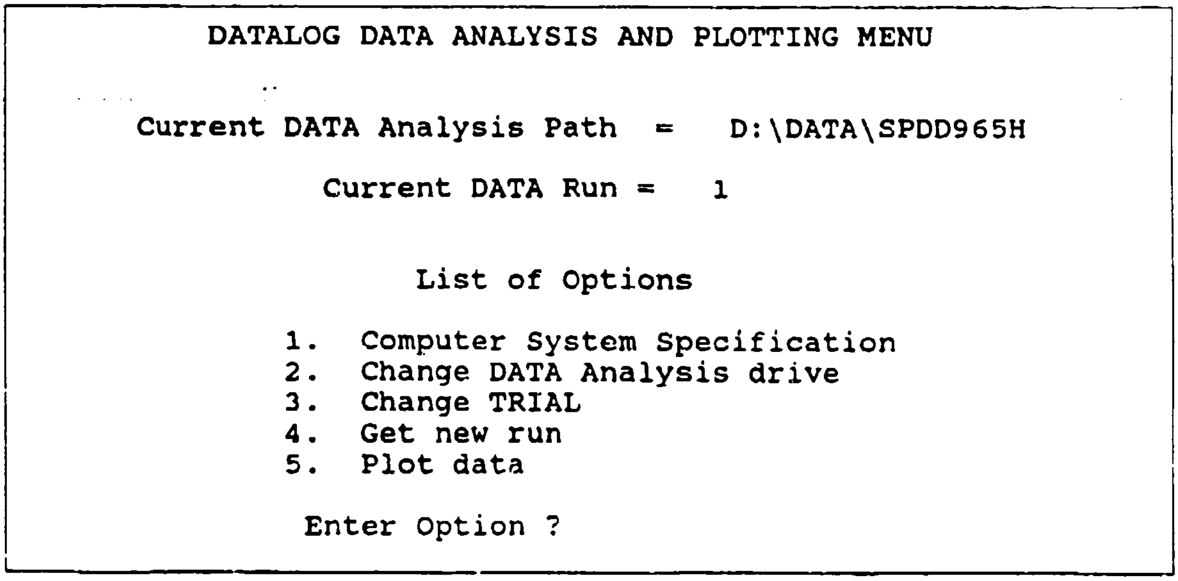
DATA ANALYSIS AND PLOTS

Choosing STH main menu (Figure 19) option 12, or PREDICT main menu (Figure 8) option 3, generates ACTH motion at a point time history plots on the terminal screen or to a printer. In either case, the plotting process and menus are the same regardless of starting point. The plot feature has two submenus, Figures 37 and 38. Tables 13 and 15, respectively, give a brief description of the submenus' options. **Data Analysis and Plots (COLLECT)**, page 25 explains the menu choices more fully.

It may be necessary to change the DATA directory or the current *TRIAL* subdirectory using options 2 or 3 to select the desired runs; however, it is not necessary for this example. Other changes to data paths are made using option 1, which modifies COMPSYS.TEX with the same procedure used to modify SMPSYS.TEX and STHSYS.TEX.

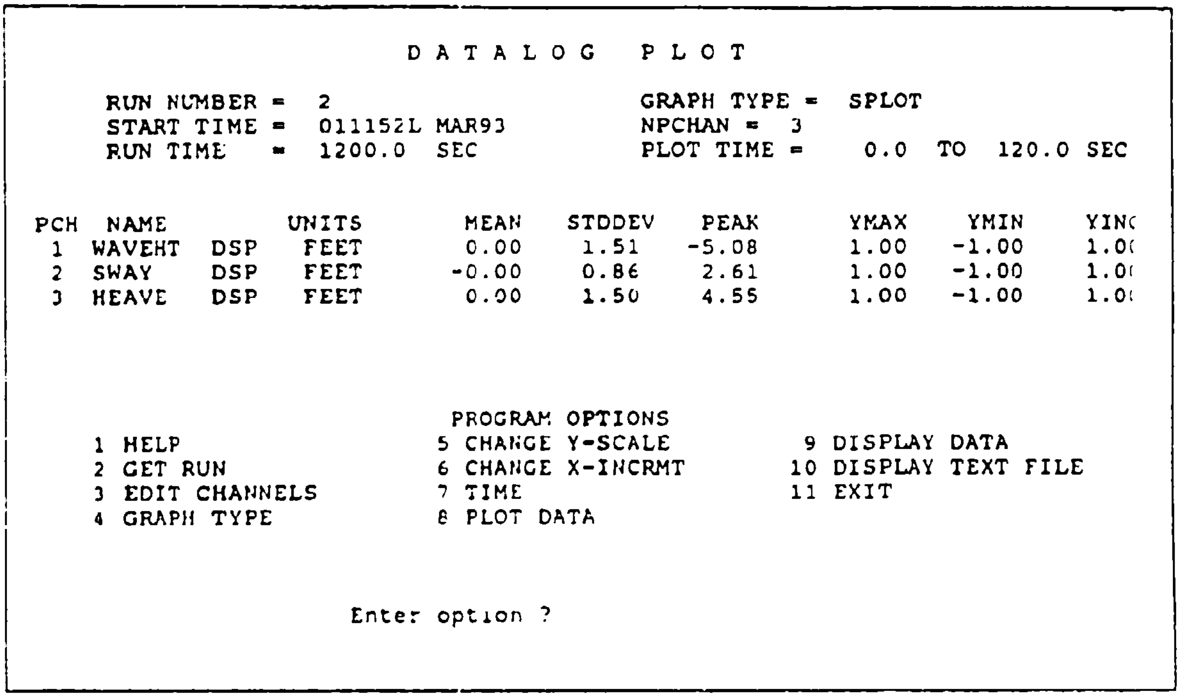
Choose submenu option 4, Figure 37, Get new run to select the run to plot from the list of available runs in the current *TRIAL* subdirectory, Figure 39. For the example,

[†] The asterisk is a wild card for the various output extensions. For ACTH, they are ASC, CON, TEX, and INT.



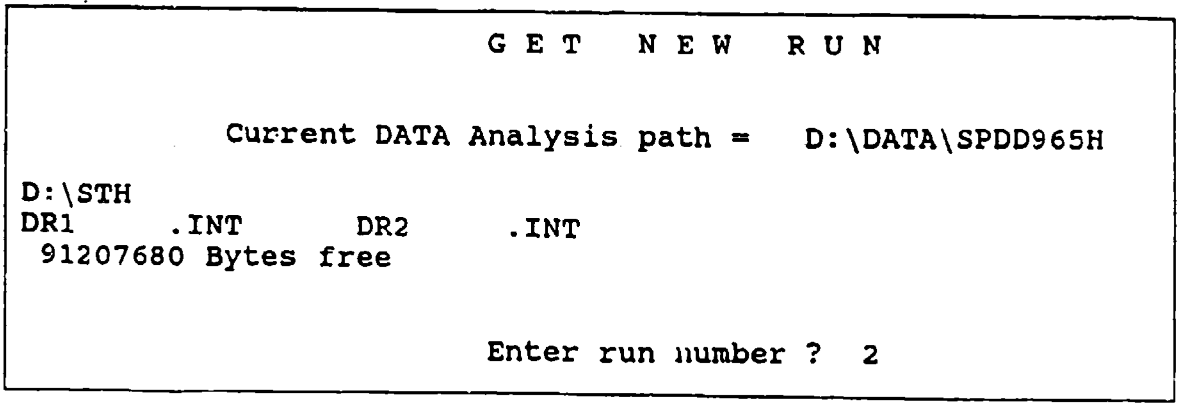
```
            DATALOG DATA ANALYSIS AND PLOTTING MENU

     Current DATA Analysis Path  =   D:\DATA\SPDD965H

                 Current DATA Run =   1


                        List of Options

              1.  Computer System Specification
              2.  Change DATA Analysis drive
              3.  Change TRIAL
              4.  Get new run
              5.  Plot data

               Enter Option ?
```

Fig. 37. CLTMAIN main menu.



```
                          D A T A L O G   P L O T

       RUN NUMBER =  2                         GRAPH TYPE =  SPLOT
       START TIME =  011152L MAR93             NPCHAN =  3
       RUN TIME   =  1200.0  SEC               PLOT TIME =    0.0  TO  120.0 SEC


  PCH  NAME          UNITS        MEAN   STDDEV    PEAK       YMAX     YMIN     YINC
   1  WAVEHT  DSP    FEET         0.00    1.51     -5.08      1.00    -1.00     1.0
   2  SWAY    DSP    FEET        -0.00    0.86      2.61      1.00    -1.00     1.0
   3  HEAVE   DSP    FEET         0.00    1.50      4.55      1.00    -1.00     1.0



                             PROGRAM OPTIONS
      1 HELP                5 CHANGE Y-SCALE        9 DISPLAY DATA
      2 GET RUN             6 CHANGE X-INCRMT      10 DISPLAY TEXT FILE
      3 EDIT CHANNELS       7 TIME                 11 EXIT
      4 GRAPH TYPE          8 PLOT DATA


                     Enter option ?
```

Fig. 38. Datalog plot menu.



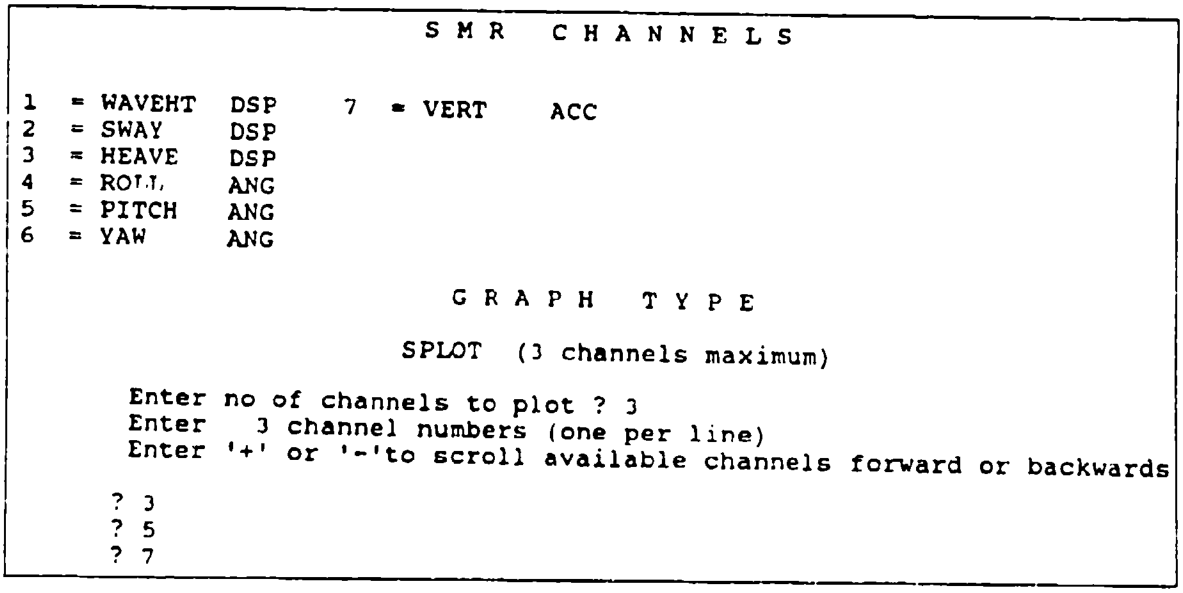
```
                          G E T    N E W    R U N

              Current DATA Analysis path =   D:\DATA\SPDD965H
D:\STH
DR1      .INT      DR2      .INT
 91207680 Bytes free


                               Enter run number ?  2
```

Fig. 39. Get new ACTH run menu.

chose the second run to plot.

To plot the data, the user selects option 5, **Plot data**. This brings up the DATALOG PLOT menu, Figure 38.

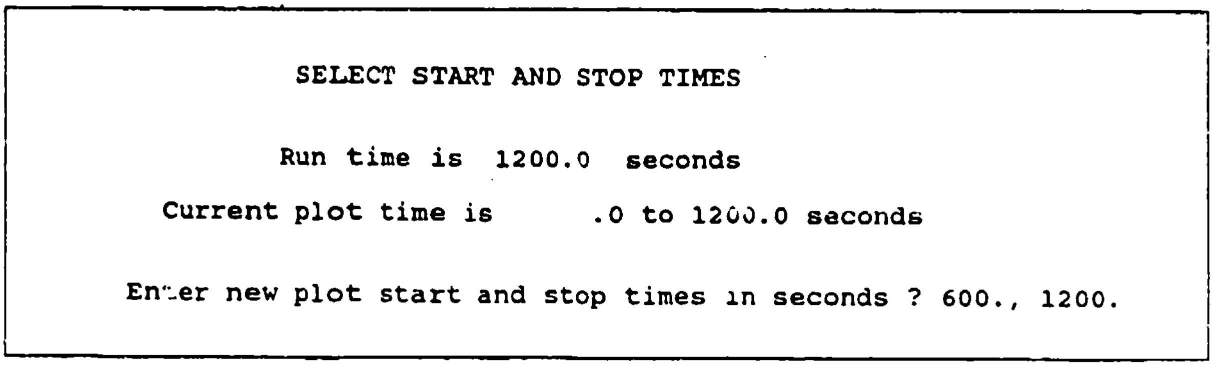
For example, plot the last 600 seconds of the wave height, heave, and boom tip vertical acceleration channels. Choose option 3, **Edit channels**, and select the desired channels from the complete channel list by typing their line numbers, Figure 40. **Time**, option 7, allows the user to enter a new begin and end time, Figure 41.



```
                              S M R   C H A N N E L S


1  = WAVEHT  DSP    7  = VERT    ACC
2  = SWAY    DSP
3  = HEAVE   DSP
4  = ROLL    ANG
5  = PITCH   ANG
6  = YAW     ANG

                               G R A P H   T Y P E

                            SPLOT  (3 channels maximum)

      Enter no of channels to plot ? 3
      Enter   3 channel numbers (one per line)
      Enter '+' or '-'to scroll available channels forward or backwards

     ? 3
     ? 5
     ? 7
```

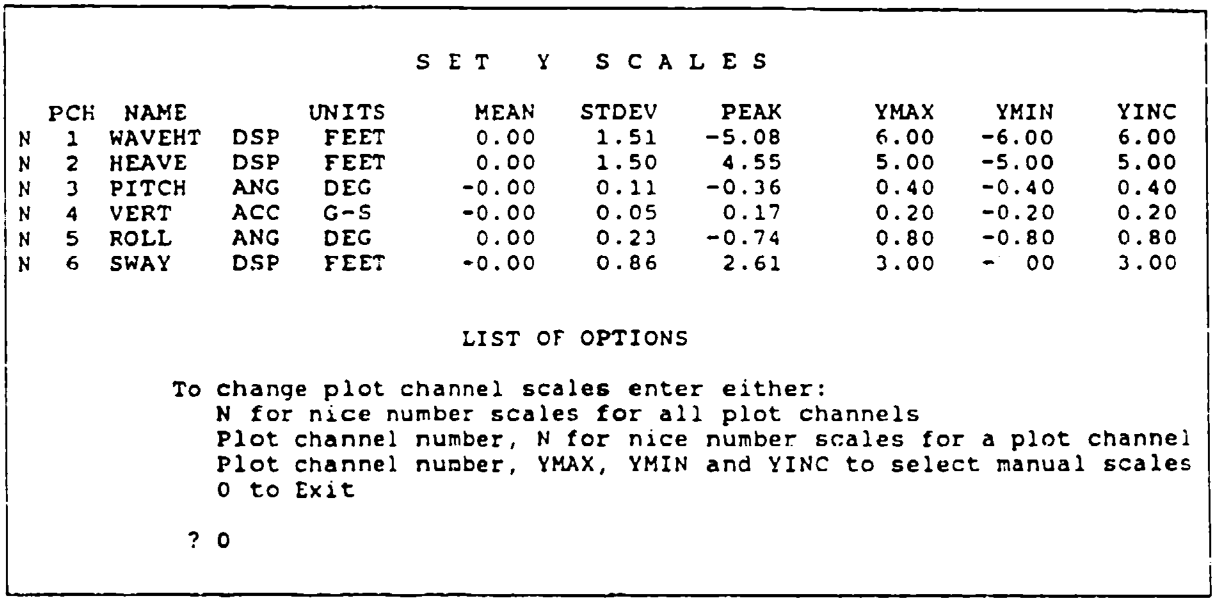
Fig. 40. Datalog plot - edit channels submenu.

After changing the time, enter 8 to PLOT DATA. Should the plotting scales be inap-



```
                 SELECT START AND STOP TIMES


                Run time is  1200.0  seconds

           Current plot time is      .0 to 1200.0 seconds


       Enter new plot start and stop times in seconds ? 600., 1200.
```

Fig. 41. Datalog plot - time subscreen.



```
                              S E T   Y   S C A L E S

   PCH  NAME          UNITS       MEAN    STDEV     PEAK        YMAX     YMIN      YINC
N   1  WAVEHT  DSP    FEET        0.00    1.51     -5.08        6.00    -6.00      6.00
N   2  HEAVE   DSP    FEET        0.00    1.50      4.55        5.00    -5.00      5.00
N   3  PITCH   ANG    DEG        -0.00    0.11     -0.36        0.40    -0.40      0.40
N   4  VERT    ACC    G-S        -0.00    0.05      0.17        0.20    -0.20      0.20
N   5  ROLL    ANG    DEG         0.00    0.23     -0.74        0.80    -0.80      0.80
N   6  SWAY    DSP    FEET       -0.00    0.86      2.61        3.00    -3.00      3.00

                                   LIST OF OPTIONS

          To change plot channel scales enter either:
             N for nice number scales for all plot channels
             Plot channel number, N for nice number scales for a plot channel
             Plot channel number, YMAX, YMIN and YINC to select manual scales
             0 to Exit

           ? 0
```

Fig. 42. Set Y scales menu.

propriate, return to the DATALOG PLOT menu and choose option 5, CHANGE Y SCALE. The CHANGE Y SCALE screen, Figure 42, has instructions detailing the various methods of scaling. Entering N (nice number scales) causes the Y scale to be autoscaled for the selected channels and is sufficient for most cases. Plotting the data again yields Figure 43. *Currently, the plotting capability is a maximum of 16 channels with 9,600 data points per channel, or 20 minutes at 8 samples per second.*

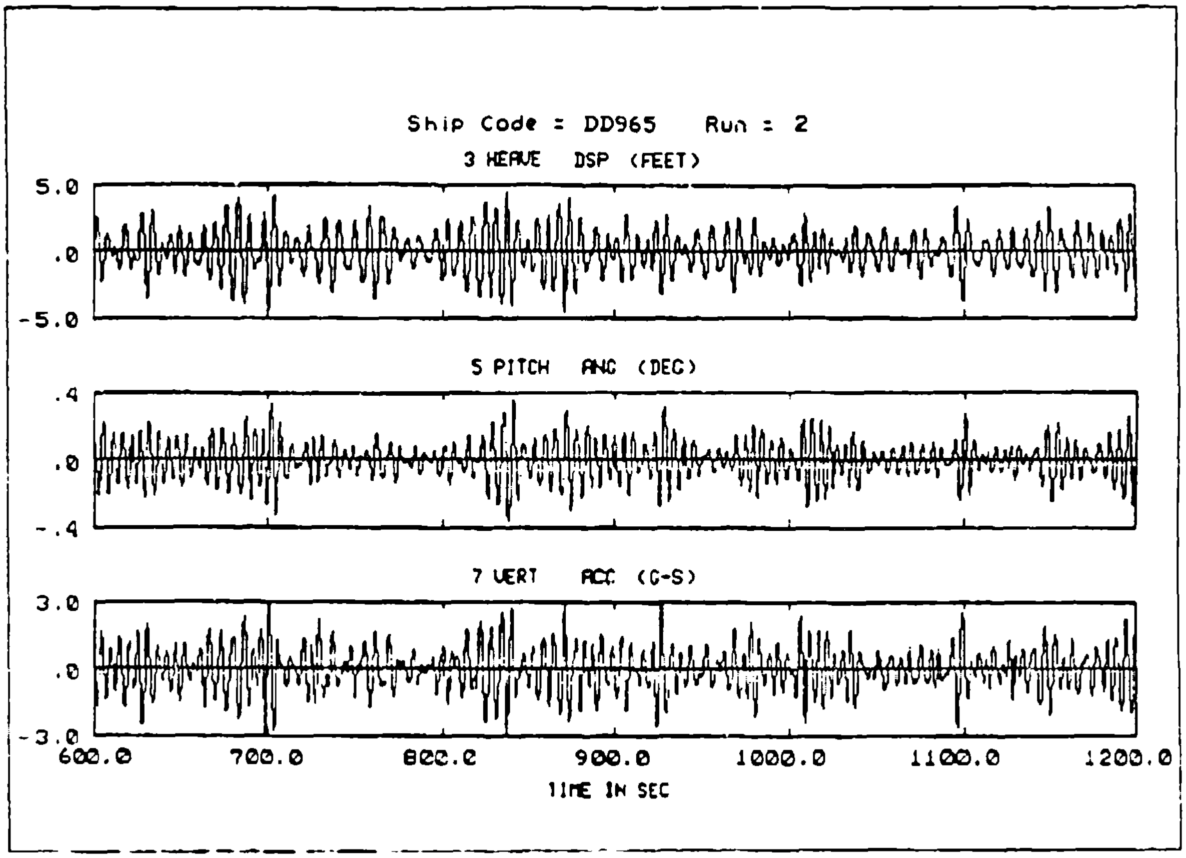


Fig. 43. Example time history plot of data.

To exit PREDICT and return to DOS, the user simply selects Exit this menu whenever possible.

CONCLUDING REMARKS

Using PREDICT, the applications manager described in this report, it is possible to generate both frequency and time domain ship motion output from a single ship input file. Absolute or relative motion at a point time histories can be developed. The time histories could be used as input, representing realistic ship motion, for other system models.

This report also provides the source code to SMP93-PC, and the appropriate list of references that fully document the Standard Ship Motion Program, the Simulated Time History Program, and the Access Time History Program.

ACKNOWLEDGMENTS

Mr. William Meyers is reponsible for most of the programming of the PC version of SMP, ACTH, and STH with help from Ms. Bennett.

APPENDIX A: STANDARD SHIP MOTION PROGRAM - PC

This appendix has a listing of the Navy Standard Ship Motion Program PC (SMP93-PC) version. The SMP version used as the baseline for the personal computer (PC) version is SMP84 which is fully documented in References 1 and 2. The theory and calculation methods are the same between the CDC, VAX, and PC versions.

The PC version is coded in Lahey Fortran and makes extensive use of overlays. The main differences between SMP84 and SMP93-PC are: an extra frequency range for response calculations, an "ORG file only" run option, and splitting the SPL file into a SPD and SPT file. Other differences involve differences between Fortran brands and in overlaying. Binary files from the CDC, VAX and PC are not interchangeable, due to differences in binary structure between machines, not due to differences between the program versions.

EXTRA FREQUENCY RANGE

The extra frequency range is especially tailored for small craft response and used when the roll period is less than or equal to nine seconds. The extra frequency set, FREQ3, was added in subroutine READ. The new frequency set is: 0.2, 0.3, 0.4, 0.5, 0.55. 0.575, 0.6, 0.625, 0.65, 0.675, 0.7, 0.725, 0.75, 0.775, 0.8, 0.825, 0.85, 0.875, 0.9, 0.95, 1.0, 1.1, 1.2, 1.3, 1.5, 1.8, 2.0, 2.5, 3.0, 3.5, and 4.0

ORG STOP AND START OPTION

With this option activated, SMP93-PC can either stop after generating the ORG file or it can start using an existing ORG file. This provides a time saving if only the ORG file is wanted. OPTN must be either 4 or 5 to start using an existing ORG file. SMPEDIT has this option available.

The flag, ORGOPTN, was added to Data Card Set 2, Program Options, as the seventh variable occupying spaces 36 - 40. Possible values are:

0 = Normal run
1 = Stop after generating ORG file
2 = Start with existing ORG file (OPTN=4 or 5)

SPLITTING SPL FILE

The unformatted SPL file was split into two files: a formatted SPT file and an unformatted SPD file. This was a minor change done to avoid memory problems. The SPT file has the speeds, headings, and response names and data. The SPD file has the RVS/T_{oe} table data.

APPENDIX B: PREDICT OVERVIEW

While PREDICT is designed to shield the user from the DOS directory and file structure, some knowledge of it is useful during installation and when choosing values for system specifications. Also understanding the file naming conventions is important to select the desired input file. The default directory names and path given here can be changed using the system specification options or editing SMPSYS.TEX, STHSYS.TEX, and/or COMPSYS.TEX.

SYSTEM REQUIREMENTS

PREDICT runs from DOS and is written in Microsoft^{TM} Fortran 5.1 and uses Halo Professional 92© for graphics capability. The memory requirements for the executables are about: 1.6 MB in SMP directory, 0.65 MB in STH directory, and 0.25 MB in COLOCT91. On a 486DX/25+ MHz machine, a 1 sea state, 7 speed SMP run takes about 40 minutes; a STH longcrested run takes about 1 minute per condition; a STH shortcrested run takes about 10 minutes per condition; and an ACTH run takes about 30 seconds per condition.

ASCII time history output files are available for use with in-house graphics programs. Appendix A documents the changes made to the CDC version of SMP84 so it would run on a personal computer.

PREDICT DIRECTORIES

BATFILE is a directory that has batch (BAT) files for the applications managers and should be added to the user's path statement in AUTOEXEC.BAT. The batch files are: PREDICT.BAT, SMPAM.BAT, STHAM.BAT, and CLTAM.BAT. They run the programs PREDICT, SMPMAIN, and STHMAIN

There are also directories for the commercial software, Fortran and Halo Professional 92© used by PREDICT. The contents of those directories depends on the software installation. Table 16 shows the default paths for the system specifications. Figure 1 shows the organization of PREDICT and the relationship between the application managers and the programs.

Table 16. Default paths for system specifications.

System specification	Default path
Halo$^{\text{TM}}$ program path	C:\HPRO_FOR
Halo$^{\text{TM}}$ graphics screen driver	IBME
Halo$^{\text{TM}}$ printer driver	EPSN
SMP program path	D:\SMP
SMP input path	D:\SMPINPUT
SMP output path	D:\SMPOUTPT
SMP data path	D:\SMPDATA
STH program path	D:\STH
STH data path	D:\STHDATA
COLLECT program path	D:\COLOCT91
COLLECT data path	D:\DATA

SMPAM DIRECTORIES

There are four SMP related directories whose default names are SMP, SMPINPUT, SMPOUTPT, and SMPDATA. See Figure 44 for a description of the SMPAM directory tree. The directory SMP has all the executables for SMPAM and a HELP subdirectory containing the on-line help files. Also in SMP are the scratch and data files for storing the current value of SMPAM variables. See Table 17 for a listing and description of these files.

SMPINPUT contains subdirectories that contain the SMP input files. Typically, these subdirectories have descriptive names such as CARRIER or BOATS, and input files fitting those descriptions are grouped together. These subdirectory names are known as the *Ship Type* in the system specifications. The input files have the *root, variant,* and *cycle number* format with an INP extension.

The output from SMP is split into the remaining two directories, SMPOUTPT and SMPDATA. Basically, SMPOUTPT has most of the ASCII data and SMPDATA has the binary data. Again, the data are actually located in subdirectories whose names reflect the input file and path used in their creation. In SMPOUTPT, the subdirectory names are the

Table 17. SMP Applications Manager scratch and data files.

File name	Description
BFILES.INP	Keeps track of which SMP output files to save or delete.
DENSITY.DAT	Previous values for generating density plot.
POLAR.DAT	Previous values for generating polar plot.
SHIPNAM.TEX	Directory listing of *Ship Type* subdirectory; only INP extensions shown.
SHIPTYP.TEX	Directory listing of SMP input path directory; only directories shown.
SMPSYS.TEX	Current path and ship for SMPAM.
TEMP.BAT	BAT file to send user back to main menu.

Ship Types and have a direct correlation to SMPINPUT. The file names are the same as in SMPINPUT except for the extension.

In SMPDATA, the subdirectory names are the *roots* of the input files. Here the file names drop the cycle number. The SMP Applications Manager (SMPAM) creates the output subdirectories automatically.

STHAM DIRECTORIES

The Simulated Time History Applications Manager has four directories, whose default names are: STH, SMPDATA, STHDATA, and DATA. The first two directories, STH and SMPDATA, contain the input; STHDATA and DATA contain the output. Figure 45 shows the default directory organization of the STHAM directories used by PREDICT. Figure 5 of Reference 3 shows another possible directory organization for use without PREDICT.

STH holds the executables for the application menu choices and various log and scratch files. See Table 18 for a list and description of the log and scratch files. SMPDATA is the same directory used by the SMPAM to store the binary SMP output used as input by STH.

Both STHDATA and DATA have the same structure and are more fully described in Reference 3, pages 15-16.22. DATA is the COLLECT data path in the system specifi-

Table 18. STH scratch and data files.

File name	Description
CONFIG	ACTH generic channel data
ERROR.TEX	ACTH run time errors
RUN.TEX	Contains run number and number of channels needed to write file configuration, CONFIG
SHIPNAM.TEX	Directory listing of *Ship Type* subdirectory; only INP extensions shown.
SHIPTYP.TEX	Directory listing of SMP input path directory; only directories shown.
STHSYS.TEX	Current paths and ship for STHAM.
STHAM.INP	Generic name for ship dependent STHAM input file.
TEMP.BAT	BAT file to send user back to main menu.

cations despite the fact it contains ACTH output. The data for various ships are in subdirectories with names based on the SMP ship origin transfer function file. STHDATA subdirectory names have the form SO*Root Variant*, e.g. SODD965H. The output file names are SR*N*.TEX and SR*N*.DAT, where *N* is the run number. Table 19 gives a list of the type of data the files contain.

DATA subdirectory names have the form SP*Root Variant*, e.g. SPDD965H. The output file names are DR*N*.INT, DR*N*.CON, DR*N*.ASC, and DR*N*.TEX, where N is the run number. Table 19 gives a list of the type of data the files contain.

PLOTTING DIRECTORIES

The plotting programs are spread between two directories SMP and COLOCT91. HULLPLOT and POLAREGA are in SMP and CLTMAIN and DLPLOT are in COLOCT91.

HULLPLOT reads the HPL for data. POLAREGA reads the SPD, SPT, POLAR.DAT, and DENSITY.DAT. The HPL, SPD, and SPT files are SMP93-PC output. POLAR.DAT and DENSITY.DAT contain the parameters from the last plot to use as defaults for the current polar or density plots.

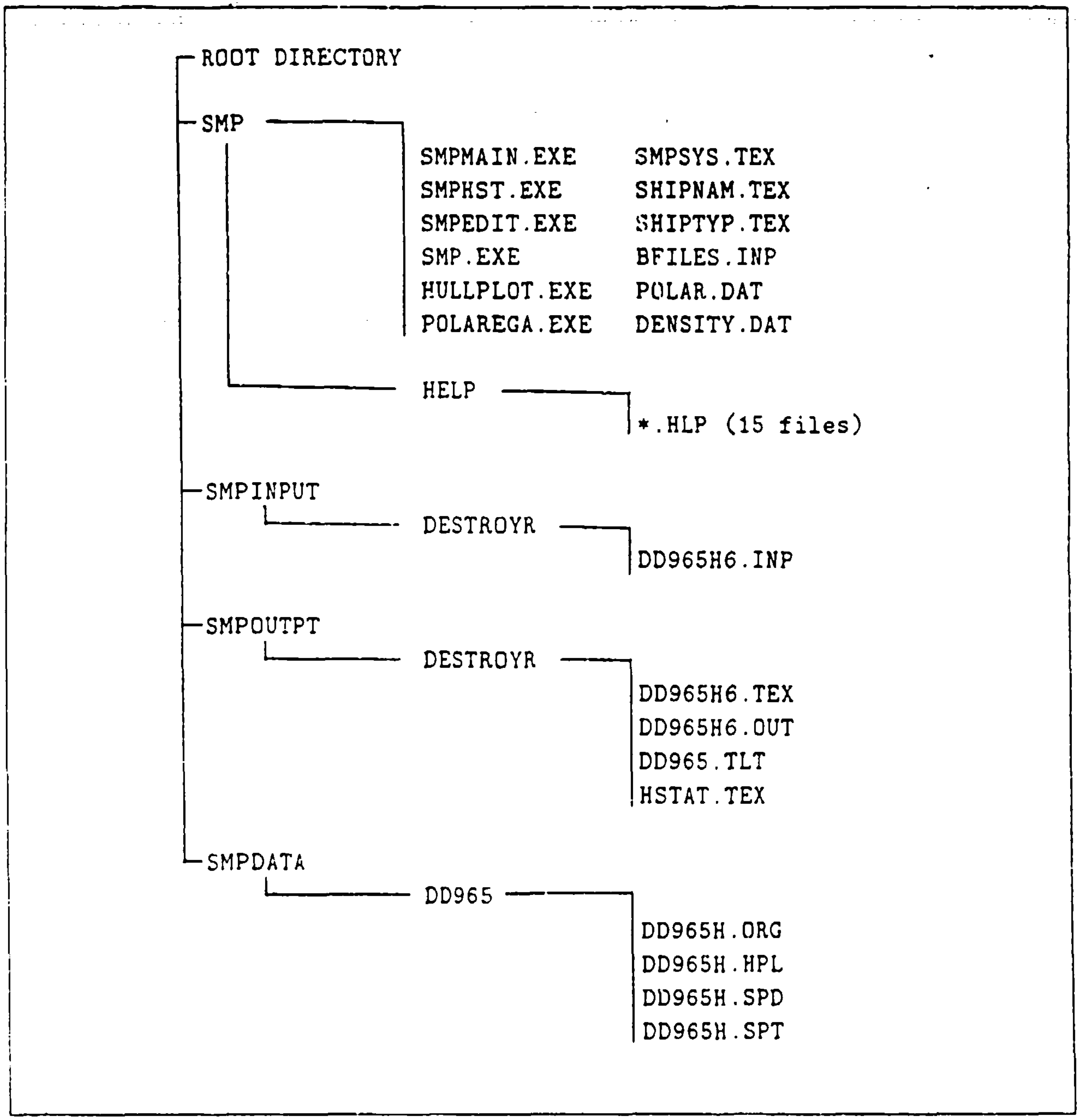


Fig. 44. SMP applications manager directory structure.

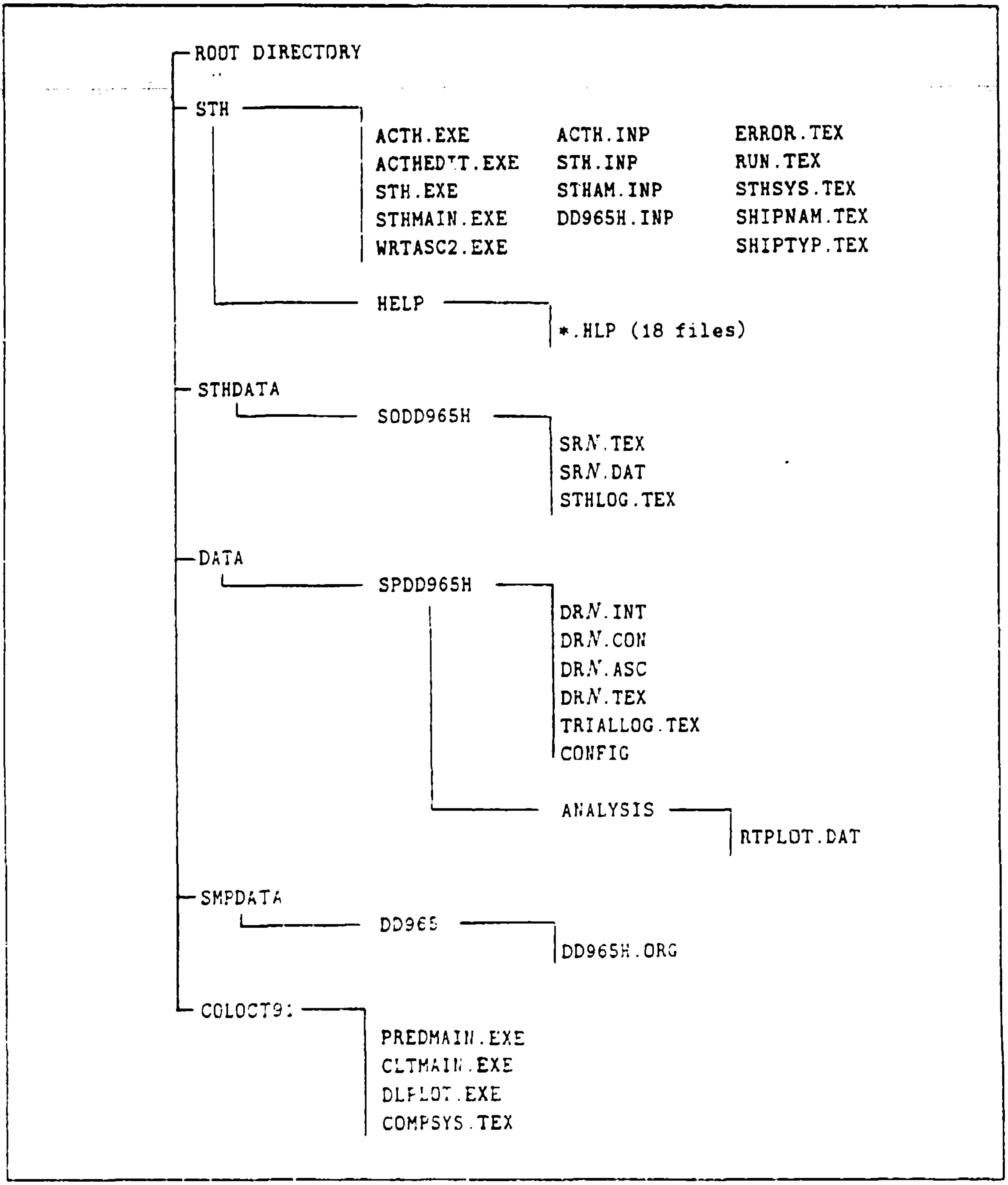


Fig. 45. STH applications manager directory structure.

Table 19. STH file names and descriptions.

File name	Description	Type
SRN.TEX	STH output echoing run parameters and screen output	ASCII
SRN.DAT	STH time history output	Binary
STHLOG.TEX	STH run log summary file	ASCII
DRN.INT	ACTH time history output	Binary
DRN.CON	ACTH file to store run dependent CONFIG data	Binary
DRN.ASC	ACTH time history output	ASCII
DRN.TEX	ACTH output echoing run parameters and screen output	ASCII
TRIALLOG.TEX	ACTH run log summary file	ASCII

CLTMAIN reads the ACTH output file, *.INT, for data. It also uses COMPSYS.TEX to determine path, directory, and current ship data. This file is also updated automatically by PREDICT when the current ship is changed. CLTMAIN is the driver for DLPLOT.

DLPLOT plots data from runs selected in CLTMAIN according to the plot definition selected within DLPLOT. RTPLOT.TEX has plot definition data from the previous plot for Datalog plots. RTPLOT.TEX is in the SP*Root Variant*\ANALYSIS subdirectory. This file and subdirectory is created automatically when entering CLTMAIN.

Printers

The graphics programs POLAREGA and DLPLOT support a limited number of printers. They are: HP LaserJet™ II and III, HP Paintjet™, HP Deskjet™, Epson, Okidata, Gemini, and IBM Proprinter™. To support other printers, modify the PARRAY array values in POLAREGA and DLPLOT, recompile, and link. Values for the printer array, PARRAY, are in the Halo™ reference manuals[6].

APPENDIX C: STHAM SHIP DEPENDENT INPUT FILE

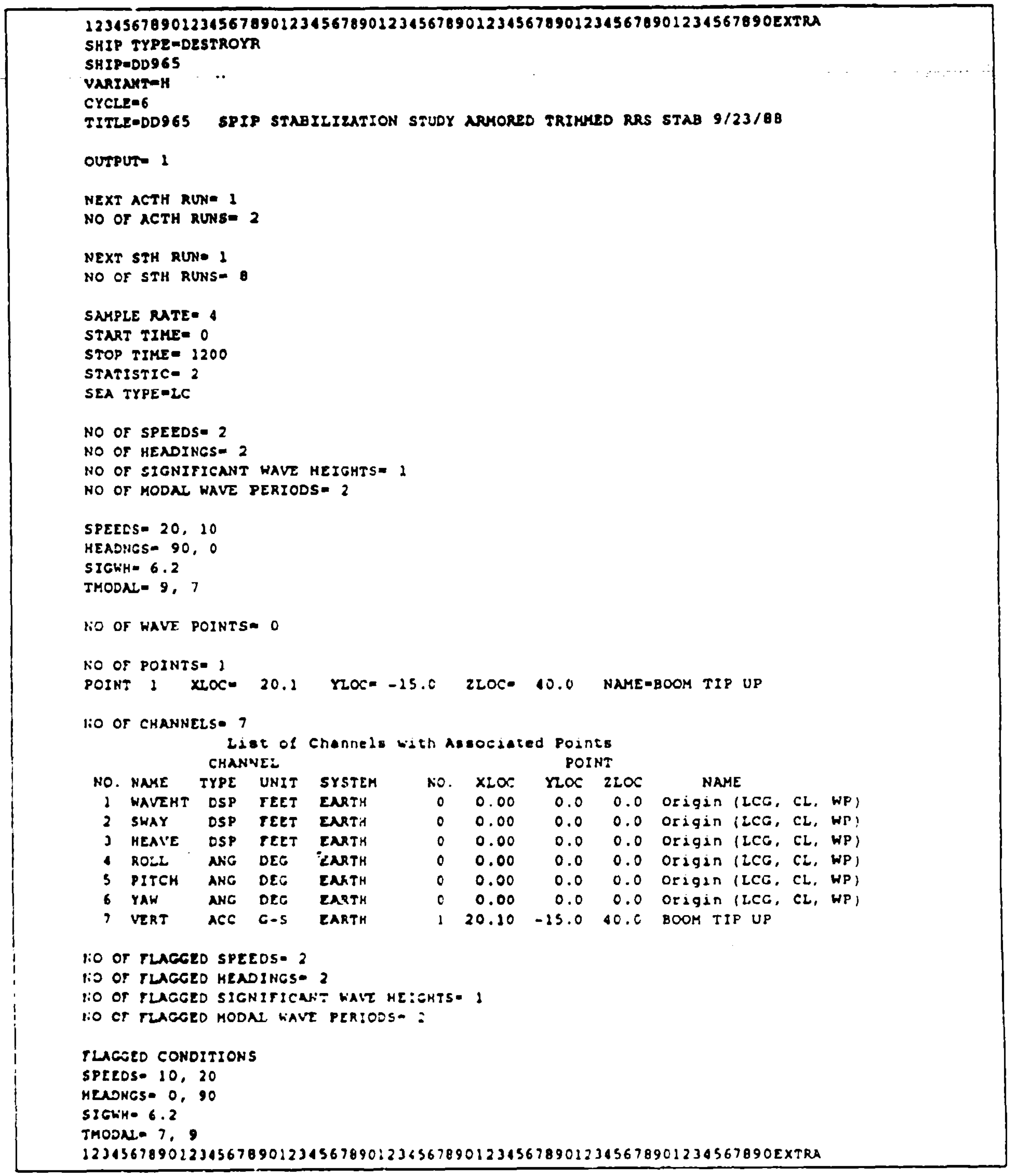
PREDICT allows switching between different ships while running, and STH and ACTH both use input from generically named input files, STH.INP and ACTH.INP. Furthermore, STH and ACTH have some, but not all, data in common. To get around this naming and data problem, PREDICT uses a ship-dependent input file that stores all the data required for STH and ACTH runs. PREDICT copies the ship dependent file to STHAM.INP and uses the paths and file names from this file as the current values, overriding the data in STHSYS.TEX. Then PREDICT simply extracts the required STH or ACTH data to use as starting points when editing or creating the generic input files, STH.INP and ACTH.INP.

The ship dependent input file name follows the *Root Variant*.INP convention, e.g. DD965H.INP. They are located in the STH data path directory, e.g. STH.

When editing STH.INP to remove unwanted sea conditions outside PREDICT, the current ship-dependent input file and STHAM.INP should also be edited to match the sea conditions in STH.INP. PREDICT updates these files automatically when edited from within PREDAM.

The format of these files is easy to understand, as each line has a commented explanation at the end of the file. The file has three main data blocks. The first gives data that PREDICT uses to find other files and to assign run numbers for STH and ACTH output files. The second block has the STH data, from sample rate to the point locations. The third block has the ACTH data, from wave points to flagged conditions. The wave and motion point data are held in common between STH and ACTH. As a rule, the flagged ACTH conditions should be a subset of the available STH conditions.

Although easy to follow, the placement of text and numbers is important. Otherwise, the file cannot be read correctly and will be useless. Figure 46 shows the format of the ship-dependent input file. The line containing numbers from 0 to 9 is not part of the file and is included in this figure as a spacing aid.



```
1234567890123456789012345678901234567890123456789012345678901234567890EXTRA
SHIP TYPE=DESTROYR
SHIP=DD965
VARIANT=H
CYCLE=6
TITLE=DD965   SPIP STABILIZATION STUDY ARMORED TRIMMED RRS STAB 9/23/88

OUTPUT= 1

NEXT ACTH RUN= 1
NO OF ACTH RUNS= 2

NEXT STH RUN= 1
NO OF STH RUNS= 8

SAMPLE RATE= 4
START TIME= 0
STOP TIME= 1200
STATISTIC= 2
SEA TYPE=LC

NO OF SPEEDS= 2
NO OF HEADINGS= 2
NO OF SIGNIFICANT WAVE HEIGHTS= 1
NO OF MODAL WAVE PERIODS= 2

SPEEDS= 20, 10
HEADNGS= 90, 0
SIGWH= 6.2
TMODAL= 9, 7

NO OF WAVE POINTS= 0

NO OF POINTS= 1
POINT  1   XLOC=   20.1   YLOC= -15.0   ZLOC=  40.0   NAME=BOOM TIP UP

NO OF CHANNELS= 7
                List of Channels with Associated Points
              CHANNEL                                POINT
 NO. NAME    TYPE  UNIT  SYSTEM      NO.   XLOC    YLOC  ZLOC       NAME
  1  WAVEHT  DSP   FEET  EARTH        0    0.00     0.0   0.0  Origin (LCG, CL, WP)
  2  SWAY    DSP   FEET  EARTH        0    0.00     0.0   0.0  Origin (LCG, CL, WP)
  3  HEAVE   DSP   FEET  EARTH        0    0.00     0.0   0.0  Origin (LCG, CL, WP)
  4  ROLL    ANG   DEG   EARTH        0    0.00     0.0   0.0  Origin (LCG, CL, WP)
  5  PITCH   ANG   DEG   EARTH        0    0.00     0.0   0.0  Origin (LCG, CL, WP)
  6  YAW     ANG   DEG   EARTH        0    0.00     0.0   0.0  Origin (LCG, CL, WP)
  7  VERT    ACC   G-S   EARTH        1   20.10   -15.0  40.0  BOOM TIP UP

NO OF FLAGGED SPEEDS= 2
NO OF FLAGGED HEADINGS= 2
NO OF FLAGGED SIGNIFICANT WAVE HEIGHTS= 1
NO OF FLAGGED MODAL WAVE PERIODS= 2

FLAGGED CONDITIONS
SPEEDS= 10, 20
HEADNGS= 0, 90
SIGWH= 6.2
TMODAL= 7, 9
1234567890123456789012345678901234567890123456789012345678901234567890EXTRA
```

Fig. 46. Ship dependent input file format for time history applications manager.

APPENDIX D: SMP93-PC SOURCE CODE LISTING

This appendix is a listing of the source code similar, but not identical, to Appendix I of Reference 1.

```
C SMP93 PROGRAM LIBRARY  -
*                          PROGRAM SMP93

*                 Standard Ship Motion Program (SMP93)
*                      for Personal Computers

*                 Operating system MS-DOS Version 4.01
*                   FORTRAN 77  using Lahey Fortran
*                    Overlay linking using PLINK86

*                     Hull plot and Speed Polar plots
*                        done in separate programs
*                      using HALO graphics language


C SUBROUTINE LIST
*---------------------------------------------------------------------

C DECK ACTFIN   -  active fins
      SUBROUTINE ACTFIN (IV,ZERO,V,OMGE,OMGE2,TAF)

      COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
     2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
     2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
     2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
     2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
     2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2)
     2,SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
     2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
     2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
     2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
     2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

      COMMON /FINCON/ IACTFN,IFCLCS,FGAIN(8),FK(3),FA(3),FB(3),
     2 FCLCS(8,2)

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
     2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
     2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
     2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
     2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
     2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
     2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
     2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
     2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
     2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
     2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
     2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
     2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
     2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
     2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
```

```
      REAL RDBLK(2692)
      EQUIVALENCE (PSUR(1),RDBLK(1))

      COMPLEX TAF(3),FGC,CTERM,ZERO

      DO 10 I=1,3
      TAF(I) = ZERO
   10 CONTINUE
      FGC = ((FK(1)-OMGE2*FK(3))+II*OMGE*FK(2))/(((FA(1)-OMGE2*FA(3))+
     2 II*OMGE*FA(2))*((FB(1)-OMGE2*FB(3))+II*OMGE*FB(2)))
      DO 30 K=1,NFNSET
      XCP = FXCP(K)
      ARM = - FMNCHD(K)/6
      YHAT = FYHAT(K)
      AP = PI*RHO*FSPAN(K)*(FMNCHD(K)/2)**2
      TEMP = FLCS(K)
      IF (IFCLCS .EQ. 1) TEMP = FCLCS(IV,K)
      FZ = (RHO/2)*FAREA(K)*TEMP
      SINGAM = SIN(FGAMMA(K)*DEGRAD)
      CTERM = FGC*(ARM*AP*OMGE2-II*OMGE*(ARM*FZ-3*AP)*V+FZ*V*V)
      M1 = 1
      IF (FNIMAG(K) .EQ. 2) M1 = 2
*    SIN(180-GAMMA)=SIN(GAMMA)  FOR FIN ON STBD SIDE
      DO 20 M=1,M1
      TAF(1) = TAF(1) - SINGAM*CTERM
      TAF(2) = TAF(2) + YHAT*CTERM
      TAF(3) = TAF(3) - SINGAM*XCP*CTERM
   20 CONTINUE
   30 CONTINUE

      RETURN
      END

C DECK ADRES
      SUBROUTINE ADRES (NL,NU,MOTV,MOTL,HJV,HJL,H7,RAO1,PHS1,RAO2,PHS2,
     2 OMEGA,NMOT,NPLANE,NOMEGA,RADDEG,COSMU,RHO,IPHS)

      COMPLEX MOTV(NMOT,NOMEGA),MOTL(NMOT,NOMEGA),HJV(NMOT,NOMEGA),
     2 HJL(NMOT,NOMEGA),H7(NOMEGA),ARES,TEMPL

      DIMENSION RAO1(NOMEGA),PHS1(NOMEGA),RAO2(NOMEGA),PHS2(NOMEGA),
     2 OMEGA(NOMEGA)

      DO 30 I=NL,NU
      DO 20 J=1,NPLANE
      ARES = H7(I)
      DO 10 N=1,NMOT
      TEMPL = MOTL(N,I)
      IF (J .EQ. 2) MOTL(N,I) = - MOTL(N,I)
      ARES = ARES + MOTV(N,I)*HJV(N,I) + MOTL(N,I)*HJL(N,I)
      MOTL(N,I) = TEMPL
   10 CONTINUE
      TEMP = - 0.5*RHO*OMEGA(I)*COSMU*AIMAG(ARES)
      IF (J .EQ. 1) RAO1(I) = TEMP
      IF (J .EQ. 2) RAO2(I) = TEMP
      IF (IPHS.EQ.1 .AND. J.EQ.1) PHS1(I) = 0.
      IF (IPHS.EQ.1 .AND. J.EQ.2) PHS2(I) = 0.
   20 CONTINUE
   30 CONTINUE

      RETURN
      END

C DECK AINPUT
      SUBROUTINE AINPUT

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
```

```
      COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
    2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
    2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
    2 LSHIPS,LTITLES
      CHARACTER*160 AS
      CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
      CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
      CHARACTER SHIPS*6,VARS*2,CYCLS*2
      INTEGER*2 OPTION

      CHARACTER*4 ALINE(20)

      FIS = SIS(1:LSIS)//'.INP'
      OPEN (UNIT=ICARD,FILE=FIS,STATUS='OLD')

      L = 0
   10 L = L + 1
      IF (MOD(L,50) .EQ. 1) WRITE (IPRIN,1000) (I,I=1,8)
 1000 FORMAT (1H1,42X,21HI N P U T   C A R D S//50X,6HCOLUMN/8X,
    2 8(9X,I1)/8H  CARD  ,8(10H1234567890)/)
      READ  (ICARD,1010) ALINE
      WRITE (IPRIN,1020) L,ALINE
 1010 FORMAT (20A4)
 1020 FORMAT (1X,I4,3X,20A4)
      IF (ALINE(1) .NE. 'STOP') GO TO 10

      CLOSE (UNI1=ICARD)

      RETURN
      END

C DECK ALAG
     FUNCTION  ALAG(X)

*  this function sets ALOG(X)=0 when x=0

     IF (X .LE. 1. E-08) GO TO 7
     ALAG=ALOG(X)
     GO TO 8
   7 ALAG=0.

   8 RETURN
     END

C DECK ALGRNG
      SUBROUTINE ALGRNG (N,W,S,AREA)

*  This subroutine computes the area under the curve for a particular
*  spectrum.  An odd number of points (frequencies) should be used.


      DIMENSION W(N),S(N)

      MN=N-2
      AREA=0.
      TEMP = O.
      DO 20 M=1,MN,2
      A=W(M+2)-W(M)
      B=W(M+2)-W(M+1)
      C=W(M+1)-W(M)
      PAREA = A*A/6.*(S(M)*(3.*C-A)/(A*C)+S(M+1)*A/(B*C)+
    2 S(M+2)*(2.*A-3.*C)/(A*B))
      TEMP = PAREA
      IF (PAREA .LT. O.) TEMP = O.
      AREA = AREA + TEMP
20    CONTINUE
      IF (MOD(N,2) .EQ. 1) GO TO 30
      DELW = W(N) - W(N-1)
      DELS = S(N) - S(N-1)
      AREA = AREA + S(N-1)*DELW + .5*DELS*DELW
   30 CONTINUE
```

```
      AREA = ABS(AREA)

      RETURN
      END

C DECK AMD
      SUBROUTINE AMD (OMEGAE,TELEM,TV,TL)

*   UNPACKS ZERO-SPEED ADDED MASS AND DAMPING AND ADDS FORWARD SPEED
*   TERMS

      COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
     2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
      LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

      COMPLEX TELEM(4,9,10)
      COMPLEX T3D(10),TV(3,3),TL(3,3)
      DIMENSION LDX(6,6)

      DATA ((LDX(I,J),J=1,6),I=1,6)
     +/  1,  0,  0,  0,  0,  0,
     +   0,  5,  0,  8,  0,  9,
     +   0,  0,  2,  0,  4,  0,
     +   0, -8,  0,  6,  0, 10,
     +   0,  0, -4,  0,  3,  0,
     +   0, -9,  0,-10,  0,  7/

      DO 20 L=1,10
      T3D(L) = (0.0,0.0)
   20 CONTINUE
      DO 40 L=LMIN,LMAX
      DO 30 K=1,4
      T3D(L) = T3D(L) + WTSI(K)*TELEM(K,ISIGMA,L)
   30 CONTINUE
   40 CONTINUE
      IF(.NOT.VRT)   GO TO 3
      DO 1  I=1,3
      IDX=2*I-1
      DO 2  J=1,3
      JDX=2*J-1
      L=LDX(IDX,JDX)
      IF(L.EQ.0)  TV(I,J)=(0.0,0.0)
      IF(L.GT.0)  TV(I,J)=T3D(L)
      IF(L.LT.0)  TV(I,J)=TV(J,I)
    2 CONTINUE
    1 CONTINUE
      TV(2,3)=TV(2,3)+V*TV(2,2)/(II*OMEGAE)
      TV(3,2)=TV(3,2)-V*TV(2,2)/(II*OMEGAE)
      TV(3,3)=TV(3,3)+V*V*TV(2,2)/OMEGAE**2
      IF(.NOT.LAT)  GO TO 6
    3 CONTINUE
      DO 4  I=1,3
      IDX=2*I
      DO 5  J=1,3
      JDX=2*J
      L=LDX(IDX,JDX)
      IF(L.EQ.0)  TL(I,J)=(0.0,0.0)
      IF(L.GT.0)  TL(I,J)=T3D(L)
      IF(L.LT.0)  TL(I,J)=TL(J,I)
    5 CONTINUE
    4 CONTINUE
      TL(1,3)=TL(1,3)-V*TL(1,1)/(II*OMEGAE)
      TL(2,3)=TL(2,3)-V*TL(2,1)/(II*OMEGAE)
      TL(3,1)=TL(3,1)+V*TL(1,1)/(II*OMEGAE)
      TL(3,2)=TL(3,2)+V*TL(1,2)/(II*OMEGAE)
      TL(3,3) = TL(3,3) + V*V*TL(1,1)/OMEGAE**2
    6 CONTINUE

      RETURN
      END

C DECK AMDPRN
      SUBROUTINE AMDPRN (PROMG,NPROMG)

*  nondimensionalizes and prints zero-speed added mass and damping

      COMMON /ENVIOR/ VK,NVK,MU,NMU.OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
    1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
    2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
    1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
    2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
    2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAFI,CROLL,
    2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
    2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
    2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
    4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
    5 IYAWRL,CHEAVE,CPITCH,CHEAFI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
    2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
    2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
    2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
    2 SPTFIL,LACFIL,LAEFIL

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
    2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
    1 RHOF,GNUS,GNUF,FTMETR

      COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
      LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

      COMMON/TELEM/TELEM
      COMPLEX TELEM(4,8,10)

      DIMENSION  LPWR(10),LDX(10)
      DIMENSION  A(10),B(10,30)
      COMPLEX T,COUM
      DIMENSION  PROMG(30)

      DATA LPWR /0,0,2,1,0,2,2,1,1,2/
      DATA LDX /1,3,5,9,2,4,6,7,8,10/

      SRGDL=SQRT(GRAV/LPP)
      LMIN=1
      IF(.NOT.VRT)  LMIN=6
      LMAX=10
      IF(.NOT.LAT)  LMAX=4
      DO 1  I=1,10
      A(I)=0.
      DO 2  J=1,NPROMG
      B(I,J)=0.
    2 CONTINUE
    1 CONTINUE
      WRITE (IPRIN,601)  TITLE
      WRITE (IPRIN,602)
      DO 3  IOMEGA=1,NPROMG
      DO 4  L=LMIN,LMAX
      LL=LDX(L)
      ASCALE=RHO*NEBLA*LPP**LPWR(L)
      BSCALE=ASCALE*SRGDL
      CALL CPLVAL (SIGMA,NSIGMA,TELEM(1,1,L),PROMG(IOMEGA),T,
     1 CDUM,IDUM)
      A(LL)=REAL(T)/(-PROMG(IOMEGA)**2)/ASCALE
      B(LL,IOMEGA)=AIMAG(T)/PROMG(IOMEGA)/BSCALE
    4 CONTINUE
      OMGND=PROMG(IOMEGA)/SRGDL
      WRITE (IPRIN,604)  OMGND,(A(L),L=1,10)
    3 CONTINUE
      WRITE (IPRIN,603)
      DO 5  IOMEGA=1,NPROMG
      OMGND=PROMG(IOMEGA)/SRGDL
      WRITE (IPRIN,604)  OMGND,(B(L,IOMEGA),L=1,10)
    5 CONTINUE
      WRITE (IPRIN,605)
  601 FORMAT (1H1,23X,20A4//42X,
     2 46HZERO-SPEED ADDED-MASS AND DAMPING COEFFICIENTS//)
  602 FORMAT (' NON-DIMENSIONAL ADDED-MASS'//
     1 ' SIGMA',3X.'A(1,1)',6X,'A(2,2)',6X,'A(3,3)',6X,'A(4,4)',6X,
     1 'A(5,5)',6X,'A(6,6)',6X,'A(2,4)',6X,'A(2,6)',6X,'A(3,5)',6X,
     1 'A(4,6)'/)
  603 FORMAT (/' NON-DIMENSIONAL DAMPING'//
     1 ' SIGMA',3X,'B(1,1)',8X,'B(2,2)',6X,'B(3,3)',6X,'B(4,4)',6X,
     1 'B(5,5)',6X,'B(6,6)',6X,'B(2,4)',6X,'B(2,6)',6X,'B(3,5)',6X,
     2 'B(4,6)'/)
  604 FORMAT (1X,F6.3,1P10E12.4)
  605 FORMAT (///' (SIGMA IS NON-DIMENSIONAL FREQUENCY)')

      RETURN
      END

C DECK ATAN2D
      FUNCTION ATAN2D (B,A,RADDEG)

*   arctangent function in degrees for any quadrant

      DATA EPS /1.E-10/

      IF (B .EQ. 0.) ATAN2D =  0.
      IF (B .GT. 0.) ATAN2D = 90.
      IF (B .LT. 0.) ATAN2D =-90.
      IF (ABS(A) .GT. EPS) ATAN2D = ATAN2(B,A)*RADDEG

      RETURN
      END

C DECK ATAN3
      FUNCTION  ATAN3(X,Y)

*  this function is to take care of the case of ATAN2(0,0)

      AX=ABS(X)
      AY=ABS(Y)
      IF(AX .LE.1.E-08 .AND.AY .LE  1. E-08) GO TO 5
      ATAN3=ATAN2(X,Y)
      GO TO 10
    5 ATAN3=0.

   10 RETURN
```

```
      END

C DECK BILGEK
      SUBROUTINE BILGEK (IBLGK)

*   calculates bilge keel damping using method of KATO
*   W. R. MCCREIGHT, DTNSRDC

      COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
     2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
     2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
     2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
     2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
     2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2)
     2,SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
     2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
     2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
     2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
     2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

      COMMON /CH3D/ ISIGMA   SMIN,SIGMAX,V,SINMU,COSMU,WTSI,
     2 IMMIN,IMMAX,IMDEL,LM   ,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     1 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTA   ,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,K  TCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      COMMON /RDGEO/ BKLEN,WBKMAX,DLBKEL(25),SRBS(25),PHIS(25),CPS(25),
     2 BKT(25),RKS(25),SSTR(25)

      COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
     2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
     2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
     2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
     2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
     2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
     2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
     2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
     2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
     2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
     2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
     2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
     2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
     2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
     2 ENSHP(8,8),RELM   .9),ITS(25),RD(25),EDDY(8,25),RGB(25)
      REAL RDBLK(2692
      EQUIVALENCE (PSUR(1),RDBLK(1))
      REAL KAPPA,KG,LAMBDA,LBKEEL
      CHARACTER*4 METER

      EXTERNAL EXP

      DATA METER /'METE'/

      LBKEEL=BKLEN
      NSM = NSTATN - 1
      DO 40 K=2,NSM
      IF (NOFSET(K) .LT. 2) GO TO 40
      IF (DLBKEL(K) .EQ. 0.) GO TO 40
      NNODES = NOFSET(K)
      R=RD(K)
      BLOCAL = 2*BMK(K)
      TLOCAL = ABS(BKT(K))
      KG = VCG + TLOCAL
      BBKEEL = BKWD(IBLGK)
      PHI=PHIS(K)
      COSPHI=CPS(K)
      RK=RKS(K)
      SS = SSTR(K)
      SRB=SRBS(K)
      RF=SRB*Y(NNODES,K)
      EPS=ATAN(SRB)
      CO=1000.*(1.44+3.8*PHI**3)
      KAPPA = R*(1.0 + RF/BLOCAL)**2 / SQRT(BLOCAL*KG/2.)
      XI=BBKEEL/(RK*PHI**0.75)
      AN=1.40+2.03*EXP(-25.*XI)
      ALPHA=2.0-AN
      CK=1.0+3.6*EXP(-9.0*KAPPA)
      SGM=2.0*BBKEEL/LBKEEL
      CN=1.98*EXP(-5.5*SGM)
      Q = (0.6*BLOCAL*TAN(PI/4. - EPS/2.) + RF - KG) * SIN(PI/4. +
     2 EPS/2.)
      PO = KG - TLOCAL/3. - 2.*RF/3.
      P1 = 0.88*(KG - TLOCAL - 0.64*(BLOCAL/2. - (TLOCAL - RF)*TAN(
     2 PI/4. + EPS/2.)))
      LAMBDA = R/(TLOCAL - RF*(BLOCAL - 2.*R)/BLOCAL)
      FLAMB=1.34*SIN(PI*LAMBDA/3.6)/
     1 (1.0+0.162*SIN(PI*(LAMBDA-0.9)/1.8))
      BCIRC = COSPHI + SS*(Q+PO-(PO-P1)*FLAMB)/(2.*BBKEEL*RK)
      DAKEEL=2.0*DLBKEL(K)*BBKEEL
      CON = 4.0*RHO/(3.0*PI)*CK*CN*BCIRC*DAKEEL*RK**3
      DO 30 IA=1,NRANG
      DO 20 IS=1,NSIGMA
      PERE = TPI/SIGMA(IS)
      F = RK*RANG(IA)*PHI**1.7/(PERE*SQRT(BBKEEL))

*    F must be in meters

      IF (PUNITS(1) .NE  METER) F = F*SQRT(FTMETR)
      CS = CO*F**(-ALPHA)/(2.68*1000.0)
      CA = 1.
      RN = (8.*BBKEEL*RK*RANG(IA) / (PERE*GNU)) * REYSCL
      IF (RN .GE. 1000.) GO TO 17
      AL10RN = ALOG(RN)/ALOG(10 )
      CA = 1.96 - 0.26*AL10RN + 0.20*SIN(PI*(AL10RN-2.19)/0.64)
   17 CONTINUE
      STADMP(IS) = CON*CS*CA*SIGMA(IS)*RANG(IA)
      STADMP(IS) = SIGMA(IS)*STADMP(IS)
      SHPDMP(IS,IA) = SHPDMP(IS,IA) + STADMP(IS)
   20 CONTINUE
   30 CONTINUE
   40 CONTINUE

      RETURN
      END

C DECK BKEDDY
      SUBROUTINE BKEDDY
      COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
     2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
     2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
     2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
     2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
     2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2)
     2,SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
     2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
     2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
     2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
     2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

      COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
     2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTECER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
     2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
     2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
     2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
     2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2) BQ(2),
     2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
     2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
     2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
     2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
     2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
     2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,SFELM(4,9),SFEELM(4,9,8),
     2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
     2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
     2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
     2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
      REAL RDBLK(2692)
      EQUIVALENCE (PSUR(1),RDBLK(1))

      DO 20 IA=1,NRANG
      ENBE(IA) = 0
      DO 10 IS=1,NSIGMA
      SHPDMP(IS,IA) = 0
   10 CONTINUE
   20 CONTINUE
      IF (NBKSET .EQ. 0) GO TO 100
      DO 30 I=1,NBKSET
      CALL CALRGM(I)
      CALL BILGEK(I)
   30 CONTINUE
      DO 40 IA=1,NRANG
      CALL SPFIT (SIGMA,SHPDMP(1,IA),BEELM(1,1,IA),NSIGMA)
      ENBE(IA) = ENCON*REVAL(BEELM(1,ISIGMA,IA),WTSI)
   40 CONTINUE
  100 CONTINUE

      RETURN
      END

C DECK BKLIFT
      SUBROUTINE BKLIFT

      COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
     2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
     2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
     2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
     2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
     2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2)
     2,SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
     2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
     2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
     2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
     2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),TITLE(20),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
     2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
     2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
     2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
     2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
     2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
     2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
     2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
     2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
     2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
     2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
     2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
     2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
     2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
     2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
      REAL RDBLK(2692)
      EQUIVALENCE (PSUR(1),RDBLK(1))

      REAL LCS,MCHORD

      IF (NBKSET .EQ. 0) GO TO 30
      EN = 0
      STASPC = LPP/20
      DO 20 K=1,NBKSET
      NBKS = NBKSTN(K)
      XBKF = LCB - BKFS(K)*STASPC
      XBKA = LCB - BKAS(K)*STASPC
      M = NBKS/2
      IF (M .EQ. 0) M = 1
      YBK = BKHB(M,K)
      ZBKF = BKWL(1,K) - (DBLWL+VCG)
      ZBKA = BKWL(NBKS,K) - (DBLWL+VCG)
      Q = 2
      SUM = 0
      DO 10 I=1,NBKS
      SUM = SUM + BKAN(I,K)
   10 CONTINUE
      GAMMA = - SUM/NBKS
      SPAN = BKWD(K)
      MCHORD = XBKF - XBKA
*    area
      AREA = SPAN*MCHORD
*    center of pressure
      XCP = XBKF - 0.5*MCHORD
      YCP = YBK + 0.5*SPAN
      ZCP = (ZBKF + ZBKA)/2
*    moment arm
      GAM = GAMMA*DEGRAD
      YHAT = YCP*COS(GAM) + ZCP*SIN(GAM)
*    effective aspect ratio
      EAR = 2*SPAN/MCHORD
*    lift curve slope
      LCS = (PI/2)*EAR
      BQ(K) = Q
      BSPAN(K) = SPAN
      BMNCHD(K) = MCHORD
      BAREA(K) = AREA
      BXCP(K) = XCP
      BYCP(K) = YCP
      BZCP(K) = ZCP
      BGAMMA(K) = GAMMA
      BYHAT(K) = YHAT
      BEAR(K) = EAR
      BLCS(K) = LCS
      EN = EN + Q*(RHO/2)*AREA*LCS*YHAT*YHAT*WPHI*ENCON
   20 CONTINUE
   30 CONTINUE
      DO 40 IV=1,NVK
      ENBL(IV) = 0
      IF (NBKSET .GT. 0) ENBL(IV) = EN*VTS(IV)
   40 CONTINUE

      RETURN
      END

C DECK BMAX
      FUNCTION  BMAX(N,X)

      DIMENSION  X(30)
      A=X(1)
      IF(N.LE.1)  GO TO 2
      DO 1  I=2,N
      IF(X(I).GT.A)  A=X(I)
    1 CONTINUE
    2 CONTINUE
      BMAX=A

      RETURN
      END

C DECK BRWVSP
      SUBROUTINE BRWVSP (NOK,SIGWH,TO,W,S)
```

```
*    this routine calculates a BRETSCHNEIDER 2-parameter wave spectrum
*            (significant  ave height, modal wave period)
*                     W.G.MEYERS, DTNSRDC, 072977

      DIMENSION W(NOK),S(NOK)

      EXTERNAL EXP

      DATA A,B /487.0626,1948.2444/
      T04 = T0**4

*     for Pierson-Moskowitz wave spectrum
*         T04 = 58.0936*SIGWH**2

      CON1 = A*SIGWH**2/T04
      CON2 = B/T04
      DO 10 I=1,NOK
      W4 = W(I)**4
      W5 = W(I)*W4
      ARG = CON2/W4
      IF (ARG.GT.50.) S(I)=0.
      IF (ARG.GT.50.) GO TO 10
      S(I) = CON1/W5*EXP(-ARG)
   10 CONTINUE

      RETURN
      END

C DECK CALRGM
      SUBROUTINE CALRGM (IBLGK)

      COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
     2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
     2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
     2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
     2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
     2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2)
     2,SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
     2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
     2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
     2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
     2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

      COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
     2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
     2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
     2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
     2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
     2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
     2 STATNM,STATIS
      CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
      INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
     2 FBNUMB,PTNUMB,ORGOPTN
      REAL KG

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      RFAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
     INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

       COMMON /RDGEO/ BKLEN,WBKMAX,DLBKEL(25),SRBS(25),PHIS(25),CPS(25),
     2 BKT(25),RKS(25),SSTR(25)

       REAL LBKEEL
       LBKEEL=0.
       NBKS = NBKSTN(IBLGK)
       STASPC = I[illegible]/20
       M = NSTATN
       NSM = NSTATN - 1
       DO 1 K=2,NSM
       M = M - 1
       IF (NOFSET(K) .LT. 2) GO TO 1
       DELTAL=0.
       SRB=0.
       PHI=0.
       COSPHI=1.
       RK=1.
       S=0.
       IF (STATN(M) .GT. BKAS(IBLGK) .OR. STATN(M) .LT. BKFS(IBLGK))
     2 GO TO 6
       IF (STATN(M+1) .GT. BKAS(IBLGK))
     2 DELTAL = (BKAS(IBLGK) - STATN(M))*STASPC
       IF (STATN(M-1) .LT. BKFS(IBLGK))
     2 DELTAL = (STATN(M) - BKFS(IBLGK))*STASPC
       IF (STATN(M+1) .LE. BKAS(IBLGK))
     2 DELTAL = DELTAL + (STATN(M+1) - STATN(M))*STASPC/2
       IF (STATN(M-1) .GE. BKFS(IBLGK))
     2 DELTAL = DELTAL + (STATN(M) - STATN(M-1))*STASPC/2
       NNODES=NOFSET(K)
       DO 10 L=1,NBKS
       IF (STATN(M) .NE. BKSTN(L,IBLGK)) GO TO 10
       RO = SQRT(BKHB(L,IBLGK)**2 + (BKWL(L,IBLGK) - (DBLWL+VCG))**2)
       ARG = BKAN(L,IBLGK)*DEGRAD
       YBKC = BKHB(L,IBLGK) + 0.5*BKWD(IBLGK)*COS(ARG)
       ZBKC = (BKWL(L,IBLGK) - DBLWL) - 0.5*BKWD(IBLGK)*SIN(ARG)
       RK = SQRT(YBKC**2 + (ZBKC-VCG)**2)
       P1 = ASIN(-VCG/RO)
       P2 = ATAN2(VCG + DBLWL - BKWL(L,IBLGK),BKHB(L,IBLGK))
       PHI = P1 + P2
       COSPHI = COS(ARG - P2)
       S=0.
       NNM=NNODES-1
       DO 3 J=1,NNM
       JS=NNODES-J+1
       IF (BKHB(L,IBLGK) .GE. Y(JS-1,K)) GO TO 4
       S=S+SQRT((Y(JS,K)-Y(JS-1,K))**2+(Z(JS,K)-Z(JS-1,K))**2)
     3 CONTINUE
     4 CONTINUE
       S = S + SQRT((Y(JS,K) - BKHB(L,IBLGK))**2 + (Z(JS,K) -
     2 (BKWL(L,IBLGK) - DBLWL))**2)

*      find minimum slope for deadrise calculation in "BILGEK"

       M2 = JS - 1
       LS = M2 - 1
       SRB = (Z(M2,K) - Z(LS,K)) / (Y(M2,K) - Y(LS,K))
       J = JS
       DO 130 I=2,M2
       J = J - 1
       JS1 = J - 1
       SLOPE = (Z(J,K) - Z(JS1,K)) / (Y(J,K) - Y(JS1,K))
       IF (SLOPE .EQ. 0.) GO TO 140
       IF (SLOPE .GT. SRB) GO TO 140
       LS= JS1
       SRB = SLOPE
   130 CONTINUE

*      extrapolate slope to centerline to get local draft
*        (excluding skeg offsets)
  140 BKT(K) = Z(LS,K) - SRB*Y(LS,K)
      IF (BKT(K) .LT. Z(1,K)) BKT(K) = Z(1,K)
      LBKEEL=LBKEEL+DELTAL
   10 CONTINUE
    6 CONTINUE
      DLBKEL(K)=DELTAL
      SRBS(K)=SRB
      PHIS(K)=PHI
      CPS(K)=COSPHI
      RKS(K)=RK
      SSTR(K)=S
    1 CONTINUE
      BKLEN=LBKEEL

      RETURN
      END

C DECK CDCOMP
      SUBROUTINE CDCOMP ( N, NDIM, A, UL, IP )

*        COMPLEX MATRIX TRIANGULARIZATION BY GAUSSIAN ELIMINATION.

*        INPUT...
*           N = ORDER OF MATRIX.
*           NDIM = DECLARED DIMENSION OF ARRAY  A .
*           A = COMPLEX MATRIX TO BE TRIANGULARIZED.

*        OUTPUT...
*           UL(I,J), I .LE. J = UPPER TRIANGULAR FACTOR, U .
*           UL(I,J), I .GT. J = MULTIPLIERS = LOWER TRIANGULAR
*                                 FACTOR, I - L .
*           IP(K), K .LT. N = INDEX OF K-TH PIVOT ROW.
*           IP(N) = (-1)**(NUMBER OF INTERCHANGES) OR 0 .

*        USE "SOLVE" TO OBTAIN SOLUTION OF LINEAR SYSTEM.
*        DETERM( A ) = IP(N)*UL(1,1)*UL(2,2)*...*UL(N,N).
*        IF IP(N) = 0, A IS SINGULAR, SOLVE WILL DIVIDE BY ZERO.

*        INTERCHANGES FINISHED IN U, ONLY PARTIALY IN L .

      REAL CABS
      COMPLEX A, UL, T
      INTEGER N, NDIM, IP, K, KP1, M, I, J
      DIMENSION A(NDIM,NDIM), UL(NDIM,NDIM)
      DIMENSION IP(NDIM)

      DO 1060 I = 1, NDIM
      DO 1000 J = 1, NDIM
      UL(J,I) = A(J,I)
 1000 CONTINUE
 1050 CONTINUE

      IP(N) = 1
      DO 1700 K = 1, N
      IF ( K .EQ. N ) GO TO 1600
      KP1 = K + 1
      M = K
      DO 1100 I = KP1, N
      IF ( CABS( UL(I,K) ) .GT. CABS( UL(M,K) ) ) M = I
 1100 CONTINUE
      IP(K) = M
      IF ( M .NE. K ) IP(N) = -IP(N)
      T = UL(M,K)
      UL(M,K) = UL(K,K)
      UL(K,K) = T
      IF ( CABS(T) .EQ. 0.0 ) GO TO 1600
      DO 1200 I = KP1, N
      UL(I,K) = -UL(I,K)/T
 1200 CONTINUE
      DO 1600 J = KP1, N
```

```
          T = UL(M,J)
          UL(M,J) = UL(K,J)
          UL(K,J) = T
          IF ( CABS(T) .EQ. 0.0 ) GO TO 1400
          DO 1300 I = KP1, N
          UL(I,J) = UL(I,J) + UL(I,K)*T
 1300     CONTINUE
 1400     CONTINUE
 1500     CONTINUE
 1600     CONTINUE
          IF ( CABS( UL(K,K) ) .EQ. 0.0 ) IP(N) = 0
 1700     CONTINUE
 99999    CONTINUE

          RETURN
          END

C DECK CEVAL
          COMPLEX FUNCTION CEVAL (CSPLNE,WEIGHT)

          COMPLEX CSPLNE(4)
          DIMENSION WEIGHT(4)

          CEVAL = (0.,0.)
          DO 10 I=1,4
          CEVAL = CEVAL + WEIGHT(I)*CSPLNE(I)
    10    CONTINUE

          RETURN
          END

C DECK CLIP
          SUBROUTINE CLIP (LIMIT,TFN,TFNMOD)

*      this routine imposes a limit on the magnitude of a dimensional
*      transfer function (surge, sway or yaw in quartering seas)
*              W.G.MEYERS, DTNSRDC, 072977

          REAL LIMIT,MAGN
          COMPLEX TFN,TFNMOD
          MAGN = CABS(TFN)
          IF (LIMIT.LE.O. .OR. MAGN.LE.LIMIT) GO TO 10

*      transfer function clipped

          RATIO = LIMIT/MAGN
          TFNMOD = RATIO*TFN
          GO TO 20
    10    CONTINUE

*      transfer function not clipped

          TFNMOD = TFN
    20    CONTINUE

          RETURN
          END

C DECK CHINR
          FUNCTION CHINR (ISKIP,AA)

          DIMENSION  AA(3,4)
          SUM=0.0
          DO 1  I1=1,4
          IF(I1.EQ.ISKIP)  GO TO 2
          I2=I1+1
          IF(I2.GT.4)  I2=1
          IF(I2.EQ.ISKIP)  I2=I2+1
          IF(I2.GT.4)  I2=1
          I3=I2+1
          IF(I3.GT.4)  I3=1
          IF(I3.EQ.ISKIP)  I3=I3+1
```

```
      IF(I3.GT.4)  I3=1
      SUM=SUM+AA(1,I1)*(AA(2,I2)*AA(3,I3)-AA(2,I3)*AA(3,I2))
    2 CONTINUE
    1 CONTINUE
      CMINR=SUM

      RETURN
      END

C DECK COFOUT
      SUBROUTINE COFOUT

*     generate coefficient file containing speed-dependant added-mass
*     and damping, exciting forces and KOCHIN functions

      COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
     2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
     2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
     2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
     2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
     2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
     2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
     2 STATNM,STATIS
      CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
      INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
     2 FBNUMB,PTNUMB,ORGOPTN
      REAL KG

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
     INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,NPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,NPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
      LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

      COMMON /STELEM/ STELEM
      COMPLEX STELEM(4,9,250)

      COMMON/TELEM/TELEM
      COMPLEX TELEM(4,9,10)

      COMMON /WGHTS/ WTDL,NORM
      REAL WTDL(10,25),NORM(4,10,25)

      COMPLEX TV(3,3),TL(3,3),EXCV(3),EXCL(3),HJV(3),HJL(3),H7
      COMPLEX STV(3,3),CDUM(3,3),SF3(25),SH3(25)
      DIMENSION SA33(25),SB33(25)

      DATA ISIGO /0/

      READ (SCRFIL) WTDL,NORM
      REWIND SCRFIL
      REWIND COFFIL
      READ (COFFIL) TELEM
      DO 300 IV=1,NVK
      V = VFS(IV)
      NMU = NNMU(IV)
      DO 200 IH=1,NMU
      HDNG = MU(IH,IV)
      SINMU = SIN(HDNG)
      COSMU = COS(HDNG)
      CON = V*COSMU/GRAV
      DO 100 IW=1,NOMEGA
      ALPHA = OMEGA(IW)*CON
      OMEGAE = ABS(OMEGA(IW)*(1.0-ALPHA))
      IF (OMEGAE .LT. SIGMA(1)) OMEGAE = SIGMA(1)
      WE = OMEGAE
      WE2 = WE*WE
      CALL FINTSP (OMEGAE)
      DO 60 K=1,NSTATN
      SA33(K) = 0.
      SB33(K) = 0.
      NPT = NOFSET(K)
      IF (NPT .LT. 2) GO TO 60
      M = (K-1)*10 + 1
      CALL AMD (OMEGAE,STELEM(1,1,M),STV,CDUM)
      SA33(K) = REAL(STV(2,2))/(-WE2)
      SB33(K) = AIMAG(STV(2,2))/WE
 60   CONTINUE
      CALL AMD (OMEGAE,TELEM,TV,TL)
      IF (ISIGMA .NE. ISIGO) CALL RDPELM
      ISIGO = ISIGMA
      CALL EXFOR (OMEGA(IW),OMEGAE,EXCV,EXCL,HJV,HJL,H7,SF3,SH3)
      WRITE (COFFIL) OMEGAE,TV,TL,EXCV,EXCL,HJV,HJL,H7
      IF (LOADS) WRITE (LCOFIL) (SF3(I),SH3(I),SA33(I),SB33(I),I=1,
     2 NSTATN)
100   CONTINUE
200   CONTINUE
300   CONTINUE
      REWIND COFFIL

      RETURN
      END

C DECK CONINT
      SUBROUTINE  CONINT (W,CELEM,NNODE)

*  SUBROUTINE TO GENERATE WEIGHTS FOR INTEGRAL ALONG CONTOUR
*  DEFINED BY PARAMETRIC SPLINE CURVE

*  INPUT
*    CELEM(8,J),J=1,(NNODE-1)  PARAMETRIC SPLINE FIT TO HULL
*    CONTOUR IN ENDPOINT-TANGENT FORMAT-
*      X(0),Y(0),DX(0),DY(0),X(1),Y(1),DX(1),DY(1)

*  OUTPUT
*    W(J),J=1,NNODE  WEIGHTS SUCH THAT INTEGRAL OF F.DS =
*    SUM OF F(J).W(J)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
       INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

       DIMENSION CELEM(8,9),FELM(4,9)
       DIMENSION  TG(10),DLDT(5)
       DIMENSION  A(5,5),          IP(5),W(10)
       DIMENSION  F(10),CF(4),CD(5),CG(8),CGI(9)
       DIMENSION  X(4),Y(4),STORCD(9,5)
       DIMENSION  SCR(3),XDS(5),YDS(5),SUM(5)

       IF (NNODE.GT.10)   WRITE (IPRIN,602) NNODE
       IF (NNODE.GT.10)   STOP
       IF (NNODE.LT.2)    WRITE (IPRIN,602) NNODE
       IF (NNODE.LT.2)    STOP
   602 FORMAT (' ERROR  -  CONIWT - NNODE = ',I5)
       NELEM=NNODE-1
       DO 1  I=1,NNODE
       TG(I)=I-1
     1 CONTINUE
       DO 2  I=1,NNODE
       W(I)=0.0
     2 CONTINUE

*   fit polynomial to dl/dt
*   set up matrices

       DO 3  I=1,5
       T=0.25*(I-1)
       A(I,1)=1.0
       DO 4 J=2,5
       A(I,J)=T*A(I,J-1)
     4 CONTINUE
     3 CONTINUE
       CALL RDCOMP (5,5,A,IP)
       IF (IP(5).EQ.0)    GO TO 101
       DO 5  K=1,NELEM
       X(1)=CELEM(1,K)
       X(2)=CELEM(3,K)
       X(3)=3.0*(CELEM(5,K)-CELEM(1,K))-2.0*CELEM(3,K)-CELEM(7,K)
       X(4)=CELEM(7,K)+CELEM(3,K)+2.0*(CELEM(1,K)-CELEM(5,K))
       Y(1)=CELEM(2,K)
       Y(2)=CELEM(4,K)
       Y(3)=3.0*(CELEM(6,K)-CELEM(2,K))-2.0*CELEM(4,K)-CELEM(8,K)
       Y(4)=CELEM(8,K)+CELEM(4,K)+2.0*(CELEM(2,K)-CELEM(6,K))

*   evaluate dl/dt at five points over (0,1)

       CALL PDER (SCR,IDXD,X,4)
       CALL PMPY (XDS,IDXDS,SCR,IDXD,SCR,IDXD)
       CALL PDER (SCR,IDYD,Y,4)
       CALL PMPY (YDS,IDYDS,SCR,IDYD,SCR,IDYD)
       CALL PADD (SUM,IDSUM,XDS,IDXDS,YDS,IDYDS)
       DO 6  I=1,5
       T=0.25*(I-1)
       CALL PVAL (TEMP,T,SUM,IDSUM)
       DLDT(I)=SQRT(TEMP)
     6 CONTINUE

*   fit polynomial to dl/dt
*   evaluate matrix solution

       CALL RSOLVE (5,5,A,DLDT,IP)
       DO 7  I=1,5
       STORCD(K,I)=DLDT(I)
     7 CONTINUE
     5 CONTINUE

*   calculate weights

       DO 8  I=1,NNODE
```

```
        DO 9  J=1,NNODE
        F(J)=0.0
    9   CONTINUE
        F(I)=1.0
        CALL SPFIT (TG,F,FELM,NNODE)
        DO 10  J=1,NELEM
        CF(1)=FELM(1,J)
        CF(2)=(FELM(3,J)-FELM(1,J)-FELM(2,J)/3.-FELM(4,J)/6.)
        CF(3)=FELM(2,J)/2.
        CF(4)=(FELM(4,J)-FELM(2,J))/6.
        DO 11  K=1,5
        CD(K)=STORCD(J,K)
   11   CONTINUE
        CALL PMPY (CG,IDG,CD,5,CF,4)
        CALL PINT (CGI,IDGI,CG,IDG)
        CALL PVAL (VAL0,0.0,CGI,IDGI)
        CALL PVAL (VAL1,1.0,CGI,IDGI)
        W(I)=W(I)+VAL1-VAL0
   10   CONTINUE
    8   CONTINUE

        RETURN
  101   CONTINUE
        WRITE (IPRIN,601) IP(5)

        STOP
  601   FORMAT (' ERROR  -  CONIWT - IP(5) = ',I5)

        END

C DECK CPFIT
        SUBROUTINE CPFIT (X, Z, CELEMS, NPTS)

•         CPFIT CREATED FROM SPFIT                    E N HUBBLE   JUNE 1977
•       FITS CUBIC NON-PARAMETRIC SPLINE SEGMENTS
•                TO SET OF COMPLEX DATA POINTS

•       INPUTS
•          X      =  ARRAY OF REAL INDEPENDENT VARIABLES
•          Z      =  ARRAY OF COMPLEX DEPENDENT VARIABLES
•          NPTS   =  NUMBER OF  (X,Z)  DATA POINTS

•       RETURN
•         CELEMS  =  ARRAY OF (NPTS-1) SEGMENTS IN FOLLOWING FORM
•                    ( (Z(I), D(I), Z(I+1), D(I+1) )      ,   WHERE
•             D   =  ARRAY OF SECOND DERIVATIVES AT DATA POINTS

•       ARRAYS A,B,C ARE MAINLY SUB DIAG., DIAGONAL, AND SUPER DIAG
•       D ARRAY IS THE RIGHT HAND SIDE OF MATRIX EQUATION
•       SECOND DERIVATIVES AT NODES ARE PLACED IN D ARRAY AFTER SOLUTION
•       SOLUTION TECHNIQUE IS GAUSSIAN ELIMINATION
•       BOUNDARY CONDITIONS SET BY EXTRAPOLATION OF SECOND DERIVATIVES

        COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
      2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RADFIL,RMSFIL,SEVFIL,SPDFIL,
      2 SPTFIL,LACFIL,LAEFIL
        INTEGER      SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
      2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RADFIL,RMSFIL,SEVFIL,SPDFIL,
      2 SPTFIL,LACFIL,LAEFIL

        COMPLEX  Z, ZDD, STORE, D, CELEMS
        DIMENSION X(NPTS),Z(NPTS),CELEMS(4,NPTS)
        DIMENSION A(100), B(100), C(100), D(100)

        N = NPTS
        NL1 = N - 1
        NL2 = N - 2
        DO 50 I=2,N
        IF (X(I) .GT. X(I-1))  GO TO 50
        WRITE (IPRIN,888) X(I-1),X(I)
        GO TO 88888
50      CONTINUE
```

```
          IF (N .LE. 100)  GO TO 100
          WRITE (IPRIN,999)
          N = 100
100       CONTINUE
          IF (N .GT. 2)  GO TO 125
          D(1) = (0.0, 0.0)
          D(2) = (0.0, 0.0)
          GO TO 375
125       CONTINUE
          IF (N .GT. 3)  GO TO 150
          ZDD = 2.*((X(3)-X(2))*Z(1)+(X(2)-X(1))*Z(3)-(X(3)-X(1))*Z(2))
         ./((X(3)-X(2))*(X(2)-X(1))*(X(3)-X(1)))
          D(1) = ZDD
          D(2) = ZDD
          D(3) = ZDD
          GO TO 375
150       CONTINUE
          DO 200 I=1,N
          A(I) = 0.0
          B(I) = 0.0
          C(I) = 0.0
          D(I) = (0.0, 0.0)
200       CONTINUE

*           set up matrices (a tridiagonal structure)

          A(1) = (X(3)-X(2))/(X(3)-X(1))
          C(1) = 2.0
          B(1) = 1.0 - A(1)
          D(1) = 6.0*((Z(3)-Z(2))/(X(3)-X(2))-(Z(2)-Z(1))/
         1 (X(2)-X(1)))/(X(3)-X(1))
          H = X(3) - X(2)
          DO 250 I=3,NL1
          HP = X(I+1) - X(I)
          C(I) = HP / (H+HP)
          B(I) = 2.0
          A(I) = 1.0 - C(I)
          D(I) = 6.0*((Z(I+1)-Z(I))/HP-(Z(I)-Z(I-1))/H)/(HP+H)
          H = HP
250       CONTINUE

*           set boundary conditions

          C(2) = (X(2)-X(1))/(X(3)-X(2))
          A(2) = 1.0
          B(2) = -1.0-C(2)
          C(2) = -A(2)*A(1)/B(1) + C(2)
          D(2) = (0.0, 0.0)
          C(N) = (X(N)-X(N-1))/(X(N-1)-X(N-2))
          A(N) = -1.0 - C(N)
          B(N) = 1.0
          D(N) = (0.0, 0.0)

*           solve equations

          II = 1
          DO 300 I=1,NL2
          I1 = I + 1
          I2 = I + 2
          AUGH = ABS (B(I))
          IF (AUGH .LT. 1.0E-06)  GO TO 275
          CONST = A(I1) / B(I)
          B(I1) = B(I1) - CONST*C(I)
          D(I1) = D(I1) - CONST*D(I)
          IF (I .NE. NL2)  GO TO 300
          A(N) = A(N) - C(N)*C(I) / B(I)
          D(N) = D(N) - C(N)*D(I) / B(I)
          GO TO 300
275       CONTINUE
          II = I + 1
          D(I) = D(I) / C(I)
          D(I1) = D(I1) - B(I1)*D(I)
```

```
      END

C DECK RLITER
      SUBROUTINE RLITER (SPINDX,TOINDX,NSPIND,NTOIND,DATA,IC,RLCALC,
    2 ROLL)

*    roll iteration

      COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
    2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
    2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLLWF(25),
    2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
    2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
    2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
    2 STATNM,STATIS
      CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
      INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
    2 FBNUMB,PTNUMB,ORGOPTN
      REAL KG

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
    1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
    2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
      INTEGER LPFIDX,LRMIDX,LSVIDX
      REAL PFIDX(235),RMIDX(183),SVIDX(3)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
    2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
    2 SPTFIL,LACFIL,LAEFIL
      INTEGER      SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
    2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
    2 SPTFIL,LACFIL,LAEFIL

      COMMON /RESPN/ NRESP,IPOINT(182),IMOTN(182),ITYPE(182),
    2 ILIN(182),ISYM(182)
      LOGICAL ILIN,ISYM

      LOGICAL LINEAR,SYMMET
      DIMENSION DATA(432),SPINDX(9),TOINDX(9),RLCALC(8,24),
    2 ROLL(13,64,4)

      IR = 1
      DO 5 N=1,NRESP
      IF (IMOTN(N).EQ.4 .AND. ITYPE(N).EQ.1) IR = N
    5 CONTINUE
      KR = IR + 1
      LINEAR = ILIN(IR)
      SYMMET = ISYM(IR)
      NPREDH = 13
      NDATA = (2 + 2*NRANG)*NPREDH
      DO 300 IS=1,NSIGWH
      K = 0
      CON = SIGWH(IS)*STATIS
      DO 200 ITO=1,NTMOD
      DO 100 IV=1,NVK
      K = K + 1
      CALL FETCH (KR,IV,ITO,DATA,RMIDX,SPINDX,TOINDX,NDATA,LRMIDX,
    2 NVK,NTMOD,RMSFIL)
      L = 2*NPREDH
      DO 10 IA=1,NRANG
      DO 10 IH=1,NPREDH
      IF (IC .EQ. 1) TEMP = DATA(L+1)
      IF (IC .EQ. 2) TEMP = DATA(L+2)
      L = L + 2
      RLCALC(IA,IH) = TEMP*CON
   10 CONTINUE
      DO 60 IH=1,NPREDH
      CALL RLITR (RLANG,NRANG,RLCALC(1,IH),ROLL(IH,K,IS))
```

```
  60  CONTINUE
 100  CONTINUE
 200  CONTINUE
 300  CONTINUE

      RETURN
      END

C DECK RLITR
      SUBROUTINE RLITR (RLANG,NRANG,RLCALC,RLANS)

      DIMENSION RLANG(8),RLCALC(8),DIFF(8),ELM(4,8)

      DO 10 IA=1,NRANG
      DIFF(IA) = RLANG(IA) - RLCALC(IA)
   10 CONTINUE
      XO = 0.
      IF (XO .GE. DIFF(1)) GO TO 20
      RLANS = RLCALC(1)
      GO TO 40
   20 IF (XO .LE. DIFF(NRANG)) GO TO 30
      RLANS = RLCALC(NRANG)
      GO TO 40
   30 CALL SPFIT (DIFF,RLANG,ELM,NRANG)
      CALL SPLVAL (DIFF,NRANG,ELM,0.,RLANS,DUM,IELM)
   40 CONTINUE

      RETURN
      END

C DECK RMS
      SUBROUTINE RMS (KREC,RAO1,RAO2,IT,N,R,B2,NPREDH,NLCH,N1,N2,DATA,
     2 IRESP,NBETA)

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      DIMENSION DATA(432),KREC(13),RAO1(30,8,13),RAO2(30,8,11),R(30),
     2 B2(36)
      REAL LMS(24)

    5 CONTINUE
      L = 2*NPREDH
      DO 60 IA=1,N
      DO 40 IH=1,NMU
      I1 = N1 + IH
      I2 = N2 - IH
      IF (I2 .LE. 0) I2 = I2 + NBETA
      IF (KREC(IH) .GT. 0) GO TO 10
      LMS(I1) = 0.
      GO TO 40
   10 DO 20 I=1,NOMEGA
   20 R(I) = RAO1(I,IA,IH)*S(I,IT)
      CALL ALGRNG (NOMEGA,OMEGA,R,LMS(I1))
      IF (KREC(IH) .EQ. 1) LMS(I2) = LMS(I1)
      IF (KREC(IH) .EQ. 1) GO TO 40
      KH = IH - 1
      DO 30 I=1,NOMEGA
   30 R(1) = RAO2(I,IA,KH)*S(I,IT)
      CALL ALGRNG (NOMEGA,OMEGA,R,LMS(I2))
   40 CONTINUE

      DO 50 IPH=1,NPREDH
      CALL XMSSC (IPH,B2,LMS,NLCH,RMSLC,RMSSC)
      IF (IRESP .EQ. 7) GO TO 45
      RMSLC = SQRT(RMSLC)
      RMSSC = SQRT(RMSSC)
   45 L = L + 1
      DATA(L) = RMSLC
      L = L + 1
      DATA(L) = RMSSC
   50 CONTINUE
   60 CONTINUE

      RETURN
      END

C DECK RMSOUT
      SUBROUTINE RMSOUT

      COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
     2 LRAOPR,ADRPR,ORGOPTN,CMNOM,KG,STATN(25),NSOFST(25),
     2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
     2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
     2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
     2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
     2 STATNM,STATIS
      CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
      INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
     2 FBNUMB,PTNUMB,ORGOPTN
      REAL KG

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD.NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8).MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),THODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV.DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25).Y(10.25).Z(10.25),FBDZV(8,10),LPP.BEAM,DBLWL,TLCB,
     2 DRAFT.LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
      INTEGER LPFIDX,LRMIDX,LSVIDX
      REAL PFIDX(236),RMIDX(183),SVIDX(3)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL.LCOFIL,ICARD.TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

      COMMON /LOADS/ NLOADS,SWGHT(25),SMASS(25),XLDSTN(10),XLDXPT(25),
     2 LSTATN(25)

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      COMMON /RESPN/ NRESP,IPOINT(182),IMOTN(182),ITYPE(182),
     2 ILIN(182),ISYM(182)
      LOGICAL ILIN,ISYM

      COMMON /SEVERE/ NRSIND,RSINDX,NSWIND,SWINDX,RSVTOE,RV,RH
      REAL RSINDX(14),SWINDX(5),RSVTOE(402)
      INTEGER RV(13),RH(13)

      COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
   2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
   2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
   2 LSHIPS,LTITLES
     CHARACTER*160 AS
     CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
     CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
     CHARACTER SHIPS*6,VARS*2,CYCLS*2
     INTEGER*2 OPTION

     DIMENSION XID(911),YID(182,5)
     DIMENSION IMODL(4),LSVRSP(13)
     CHARACTER*4 RTITL(2),RTYPE(3),RUNIT(3),RSPNME(2,13)
     CHARACTER*1 BLANK,BT(80)
     CHARACTER*2 AC(2),AT,AVK
     CHARACTER*10 PARS1
     CHARACTER*110 PARS,SEA
     CHARACTER*100 PARS2
     DIMENSION HDNG(24)
     DIMENSION DATA(432),SPINDX(9),TOINDX(9),RMS(8,24),ROLL(13,64,4),
   2 ELM(4,8),RMSTBL(25,8,8),TOETBL(25,8,8),TEMRMS(13),TEMTOE(13)
     DIMENSION INDXRL(25),INDXHD(25),HEADNG(26)
     LOGICAL LINEAR,SYMMET
     INTEGER TOETBL,HEADNG,TEMTOE
     EQUIVALENCE (IPOINT,YID),(XID,MRESP)
     CHARACTER*4 METER,MET,FT,ACOND(3,2),BS,SUNIT

     DATA METER,MET,FT /'METE',' M, ',' FT,'/
     DATA LSVRSP /3,5,2,4,6,9,8,9,8,9,8,9,8/
     DATA RSPNME /'HEAV','E','PITC','H',' SWA','Y',' ROL','L',
   2   '  YA','W','P1VA','C','P1LA','C','P2VA','C','P2LA','C',
   2   'P3VA','C','P3LA','C','P4VA','C','P4LA','C'/
     DATA INDXRL /1,2,3,4,5,6,7,8,9,10,11,12,13,12,11,10,9,8,7,6,5,4,
   2 3,2,1/
     DATA INDXHD /13,12,11,10,9,8,7,6,5,4,3,2,1,24,23,22,21,20,19,18,
   2 17,16,15,14,13/
     DATA HEADNG /0,15,30,45,60,75,90,105,120,135,150,165,180,
   2 360,345,330,315,300,285,270,255,240,225,210,195,180/
     DATA HDNG /0.,15.,30.,45.,60.,75.,90.,105.,120.,135.,150.,165.,
   2 180.,195.,210.,225.,240.,255.,270.,285.,300.,315.,330.,345./
     DATA AC /'LC','SC'/
     DATA BLANK /' '/
     DATA ACOND /'LONG','CRES','TED ','SHOR','TCRE','STED'/

     NSVRSP = 13
     NHEAD = 24
     NSPIND = NVK + 1
     NTOIND = NTMOD + 1
     NID = 911

     FIS = SDS(1:LSDS)//'.RMS'
     OPEN (UNIT=RMSFIL,FILE=FIS,STATUS='UNKNOWN',
    2 ACCESS='DIRECT',RECL=1750)

*           modified to run on VAX/VMS
*   CDC     CALL READMS (RMSFIL,XID,NID,1)

     READ (RMSFIL,REC=1) (XID(I),I=1,432)
     READ (RMSFIL,REC=2) (XID(I),I=433,796),L
     IF (L .EQ. NID) READ (RMSFIL,REC=3) (XID(I),I=797,911)

*   CDC     L = LENGTH(RMSFIL)

     M = (L-1)/5
     IF(L .NE. NID) M = 159

*   M = 159  means RMSFIL was generated by SMP81
*   M = 182  means RMSFIL was generated by SMP84

     K = 1
     DO 750 J=1,5
     DO 750 I=1,M
     K = K + 1
```

```
      YID(I,J) = XID(K)
  750 CONTINUE
      NRESP = MRESP
      DO 770 IS=1,NSIGWH
*    find most probable period
      SWH = SIGWH(IS)
      IF (PUNITS(1) .NE. METER) SWH = SWH*FTMETR
*    significant wave height ranges below are in meters
*    sea state 1
      IF (SWH .LE. 0.59) PER = 5.0
*    sea state 2
      IF (SWH.GT.0.59 .AND. SWH.LE.1.26) PER = 5.0
*    sea state 3
      IF (SWH.GT.1.26 .AND. SWH.LE.1.73) PER = 7.0
*    sea state 4
      IF (SWH.GT.1.73 .AND. SWH.LE.2.24) PER = 7 0
*    sea state 5
      IF (SWH.GT.2.24 .AND. SWH.LE.3.97) PER = 9.0
*    sea state 6
      IF (SWH.GT.3.97 .AND. SWH.LE.6.34) PER = 11.0
*    sea state 7
      IF (SWH.GT.6.34 .AND. SWH.LE.12.29) PER = 15.0
*    sea state 8
      IF (SWH.GT.12.29 .AND. SWH.LE.18.77) PER = 19.0
*    greater than sea state 8
      IF (SWH .GT. 18.77) PER = 19.0
      IF (PER .LT. TMODAL(1)) PER = TMODAL(1)
      IF (PER .GT. TMODAL(NTMOD)) PER = TMODAL(NTMOD)
      IMODL(IS) = 1
      DO 760 LT=1,NTMOD
      IF (ABS(PER-TMODAL(LT)) .LT. 0.0001) IMODL(IS) = LT
  760 CONTINUE
  770 CONTINUE
      ISKPSV = 0
      IF (IMOTN(1) .NE. 1) ISKPSV = 1
*    ISKPSV = 0  all motions - output severe motion tables
*    ISKPSV = 1  roll motion only - skip severe motion tables
      IF (ISKPSV .EQ. 1) GO TO 820
      FIS = SDS(1:LSDS)//'.SEV'
      OPEN (UNIT=SEVFIL,FILE=FIS,STATUS='UNKNOWN',
    2 ACCESS='DIRECT',RECL=1620)
      NSVRSP = 5 + 2*NPTLOC
      IF (NSVRSP .GT. 13) NSVRSP = 13
      CALL SETSEV (NSVRSP,LSVRSP)
      NRSIND = NSVRSP + 1
      NSWIND = NSIGWH + 1
      NRECD = 0
  820  CONTINUE

       L = - 3
       DO 2 I=1,20
       L = L + 4
       K = L + 3
       READ (TITLE(I),5000) (BT(J),J=L,K)
 5000  FORMAT (4A1)
    2  CONTINUE
       L = 0
       DO 4 I=1,80
       L = L + 1
       IF (BT(I) .NE. BLANK) GO TO 6
    4  CONTINUE
    6  CONTINUE
       IF (L.EQ.60 .AND. BT(80).EQ.BLANK) L = 1
       M = L + 9
       IF (M .GT. 80) M = 80
       WRITE (PARS1,5010) (BT(I),I=L,M)
 5010  FORMAT (10A1)
       WRITE (PARS2,5020) TITLE
 5020  FORMAT (20A4,20X)

*    write to speed polar data and text files

       FIS = SDS(1:LSDS)//'.SPD'
       OPEN(SPDFIL,FILE=FIS,ACCESS='DIRECT',STATUS='UNKNOWN',
     2 FORM='UNFORMATTED',RECL=768)

       FIS = SDS(1:LSDS)//'.SPT'
       OPEN(SPTFIL,FILE=FIS,STATUS='UNKNOWN')

       WRITE (SPTFIL,5022) PARS1,PARS2
 5022  FORMAT(A10/A100)

       PRIDIR = 90.
       SECDIR = 0.

       WRITE (SPTFIL,5023) NVK,NHEAD
 5023  FORMAT(2I5)

       WRITE (SPTFIL,5024) (VK(IV),IV=1,NVK)
       WRITE (SPTFIL,5024) (HDNG(IH),IH=1,NHEAD)
 5024  FORMAT(8F10.4)

*    loop over longcrested, shortcrested waves

       DO 500 IC=1,2
       CALL RLITER (SPINDX,TOINDX,NSPIND,NTOIND,DATA,IC,RMS,ROLL)

*          change for VAX/VMS version
*  CDC     CALL STINDX (SEVFIL,RSINDX,NRSIND)
*  CDC     DO 7 I=1,NRSIND
*  CDC     RSINDX(I) = 0.
*  CDC 7   CONTINUE


*    loop over response

       DO 400 IR=1,NRESP
       JR = 0
       IF (ISKPSV .EQ. 1) GO TO 19
       DO 18 LR=1,NSVRSP
       IF (IR .NE. LSVRSP(LR)) GO TO 18
       JR = LR
       GO TO 19
   18  CONTINUE
   19  CONTINUE
       KR = IR + 1
       IP = IPOINT(IR)
       IM = IMOTN(IR)
     IT = ITYPE(IR)
     CALL RSTITL (IP,IM,IT,RTITL,RTYPE,RUNIT,PARS)
     LINEAR = ILIN(IR)
     SYMMET = ISYM(IR)
     NPREDH = 13
     IF (.NOT. SYMMET) NPREDH = 24
     N = 1
     IF (.NOT. LINEAR) N = NRANG
     NDATA = (2 + 2*N)*NPREDH

*    change for VAX/VMS version
*  CDC   IF (JR .EQ. 0) GO TO 21
*  CDC   CALL STINDX (SEVFIL,SWINDX,NSWIND)
*  CDC   DO 8 I=1,NSWIND
*  CDC   SWINDX(I) = 0.
*  CDC 8 CONTINUE
*  CDC21 CONTINUE

*   loop over significant wave height

     DO 300 IS=1,NSIGWH
     CON = SIGWH(IS)*STATIS
     IF (IM.EQ.15) CON - SIGWH(IS)

*   loop over modal wave period

     K = 0
     DO 200 ITO=1,NTMOD
     SWHMAX = .202*TMODAL(ITO)**2
     IF (PUNITS(1) .EQ. METER) SWHMAX = SWHMAX*FTMETR

*   loop over speed

     DO 100 IV=1,NVK
     K = K + 1
     IF (SIGWH(IS) .GT. SWHMAX) GO TO 100
     CALL FETCH (KR,IV,ITO,DATA,RMIDX,SPINDX,TOINDX,NDATA,LRMIDX,
   2 NVK,NTMOD,RMSFIL)

*   loop over heading

     L = 2*NPREDH
     DO 10 IA=1,N
     DO 10 IH=1,NPREDH
     IF (IC .EQ. 1) TEMP = DATA(L+1)
     IF (IC .EQ. 2) TEMP = DATA(L+2)
     L = L + 2
     RMS(IA,IH) = TEMP*CON
  10 CONTINUE
     N1 = NHEAD + 1
     DO 60 IH=1,N1
     IF (IH .GT. NPREDH) GO TO 50
     LH = INDXHD(IH)
     JC = (IH-1)*2 + IC
     IF (.NOT. LINEAR) GO TO 20
     RMSTBL(LH,ITO,IV) = RMS(1,IH)
     GO TO 40
  20 KH = INDXRL(IH)
     RLCALC = ROLL(KH,K,IS)
     IF (RLCALC .GE. RLANG(1)) GO TO 30
     RMSTBL(LH,ITO,IV) = RMS(1,IH)
     GO TO 40
  30 IF (RLCALC .LE. RLANG(NRANG)) GO TO 35
     RMSTBL(LH,ITO,IV) = RMS(NRANG,IH)
     GO TO 40
  35 CALL SPFIT (RLANG,RMS(1,IH),ELM,NRANG)
     CALL SPLVAL (RLANG,NRANG,ELM,RLCALC,RMSTBL(LH,ITO,IV),DUM,IELM)
  40 TOETBL(LH,ITO,IV) = DATA(JC) + .5001
     GO TO 60
  50 JH = INDXRL(IH)
     RMSTBL(IH,ITO,IV) = RMSTBL(JH,ITO,IV)
     TOETBL(IH,ITO,IV) = TOETBL(JH,ITO,IV)
  60   CONTINUE
 100   CONTINUE
       IF (SIGWH(IS) .GT. SWEMAX) GO TO 200
       TOEMIN = 99.0
       TOEMAX = 0.0
       RMSMIN = RMSTBL(1,ITO,1)
       RMSMAX = RMSMIN
       DO 120 IV=1,NVK
       DO 110 IH=1,NHEAD
       TEMP = RMSTBL(IH,ITO,IV)
       VTMP = TOETBL(IH,ITO,IV)
       IF (VTMP .GT. 99.) VTMP = 99.
       IF (TEMP .LT. RMSMIN) RMSMIN = TEMP
       IF (VTMP .LT. TOEMIN) TOEMIN = VTMP
       IF (VTMP .GT. TOEMAX) TOEMAX = VTMP
       IF (TEMP .LT. RMSMAX) GO TO 110
       RMSMAX = TEMP
       IF (JR .EQ. 0) GO TO 110
       IF (ITO .NE. IMODL(IS)) GO TO 110
       IF (SYMMET .AND. IH.GT.13) GO TO 110
       MXV = IV
       MXH = IH
 110   CONTINUE
 120   CONTINUE
       IF (JR .EQ. 0) GO TO 150
       IF (ITO .NE. IMODL(IS)) GO TO 150
       RSVTOE(1) = MXV
       RSVTOE(2) = MXH
       IE = 2
       DO 130 IV=1,NVK
       DO 130 IH=1,NHEAD
       IE = IE + 1
       RSVTOE(IE) = RMSTBL(IH,ITO,IV)
       IE = IE + 1
       RSVTOE(IE) = TOETBL(IH,ITO,IV)
 130   CONTINUE

*    write to severe motion file

*        change for VAX/VMS version
*  CDC     CALL WRITMS (SEVFIL,RSVTOE,IE,IS)

       NRECD = NRECD + 1
       WRITE (SEVFIL,REC=NRECD) RSVTOE

 150   CONTINUE

*    write to speed polar file

       ISIGWH = SIGWH(IS)*100.
       IF (ISIGWH .GE. 1000) WRITE (BS,3001) ISIGWH
       IF (ISIGWH .LT. 1000) WRITE (BS,3002) ISIGWH
       IF (ISIGWH .LT.  100) WRITE (BS,3003) ISIGWH
       IF (ISIGWH .LT.   10) WRITE (BS,3004) ISIGWH
3001   FORMAT (I4)
3002   FORMAT (1H0,I3)
3003   FORMAT (2H00,I2)
3004   FORMAT (3H000,I1)
3000   FORMAT (1H0,I1)
3010   FORMAT (I2)
       ITMODL = TMODAL(ITO) + .5
       IF (ITMODL .LT. 10) WRITE (AT,3000) ITMODL
       IF (ITMODL .GE. 10) WRITE (AT,3010) ITMODL
       SUNIT = MET
       IF (PUNITS(1) .NE. METER) SUNIT = FT
       WRITE (SEA,3020) BS,AT,AC(IC),SIGWH(IS),SUNIT,TMODAL(ITO),
     2 (ACOND(I,IC),I=1,3),(STATNM(I),I=1,3)
3020   FORMAT (2HBR,A4,2A2,32H BRETSCHNEIDER SEAWAY -  SIGWH =,F6.2,A4,
     2 10H  TMODAL =,F6.2,7H SEC,  ,3A4,4X,3A4,7X)
       WRITE (SPTFIL,5025) PARS,SEA
5025   FORMAT(A110)
       WRITE (SPTFIL,5026) RMSMIN,RMSMAX,TOEMIN,TOEMAX
 5026  FORMAT(4F10.5)
       WRITE (SPDFIL) ((RMSTBL(IH,ITO,IV),IV=1,NVK),IH=1,NHEAD)
       WRITE (SPDFIL) ((TOETBL(IH,ITO,IV),IV=1,NVK),IH=1,NHEAD)
  200  CONTINUE
       IF (IT .GT. 1 .AND. VLACPR.EQ.0) GO TO 300

*    print RMS/TOE tables

       DO 250 IPAGE=1,2
       IF (IPAGE.EQ.2 .AND. SYMMET) GO TO 250
       WRITE (IPRIN,1000) TITLE
 1000  FORMAT (1H1,22X,20A4)
       IF (IC .EQ. 1) WRITE (IPRIN,1010)
       IF (IC .EQ. 2) WRITE (IPRIN,1020)
 1010  FORMAT (/58X,11HLONGCRESTED)
 1020  FORMAT (/58X,12HSHORTCRESTED)
       IF (PUNITS(1) .NE. METER) WRITE (IPRIN,1030) SIGWH(IS)
 1030  FORMAT (45X,25HSIGNIFICANT WAVE HEIGHT =,F6.2,5H FEET)
       IF (PUNITS(1) .EQ. METER) WRITE (IPRIN,1031) SIGWH(IS)
 1031  FORMAT (45X,25HSIGNIFICANT WAVE HEIGHT =,F6.2,7H METERS)
       IF (IP.GT.0 .AND. IM.LE.3) WRITE (IPRIN,1032) (PTNAME(I,IP),
     2 I=1,8),XPTLOC(IP),YPTLOC(IP),ZPTLOC(IP)
       IF (IP.GT.0 .AND. IM.EQ.15) WRITE (IPRIN,1032) (PTNAME(I,IP),
     2 I=1,8),XPTLOC(IP),YPTLOC(IP),ZPTLOC(IP)
 1032  FORMAT (/27X,8A4,2X,5HXFP =,F7.2,2X,5HYCL =,F7.2,2X,5HZBL =,F7.2)
       IF (IP.GT.0 .AND. IM.EQ.8) WRITE (IPRIN,1033) (FBNAME(I,IP),
     2 I=1,5),XPTFBD(IP),YPTFBD(IP),ZPTFBD(IP)
 1033  FORMAT (/33X,5A4,2X,5HXFP =,F7.2,2X,5HYCL =,F7.2,2X,5HZBL =,F7.2)
       IF (IP.GT.0 .AND. (IM.GE.10 .AND. IM.LE.14)) WRITE (IPRIN,1073)
     2 XLDSTN(IP)
 1073  FORMAT (/58X,7HSTATION,F5.1)
       IF (IM.NE.15) WRITE (IPRIN,1034) RTITL,RTYPE,RUNIT
 1034  FORMAT (/54X,2A4,1X,3A4/58X,3A4)
       IF (IM.EQ.15) WRITE (IPRIN,1035)
 1035 FORMAT(/50X,26HHORIZONTAL FORCE ESTIMATOR/58X,4H (G))
       IF (IM.LT.4 .AND. IT.EQ.3) WRITE (IPRIN,1036)
       IF (IM.EQ.15) WRITE (IPRIN,1036)
 1036  FORMAT (58X,12H(ACC. X 100))
       IF (IP.GT.0 .AND. (IM.GE.10 .AND. IM.LE.11)) WRITE (IPRIN,1063)
 1063  FORMAT (/57X,14H(FORCE / 100 ))
       IF (IP.GT.0 .AND. (IM.GE.12 .AND. IM.LE.14)) WRITE (IPRIN,1065)
 1065  FORMAT (/54X,16H(MOMENT / 10000))
       IF (IM .EQ.7) WRITE (IPRIN,1038)
 1038  FORMAT (57X,14H(FORCE / 1000))
       IF (IM.NE.15) WRITE (IPRIN,1040)(STATNM(I),I=1,3)
       IF (IM.EQ.15) WRITE (IPRIN,1041)
 1040  FORMAT (/40X,3A4,39H VALUE / ENCOUNTERED MODAL PERIOD (TOE))
 1041  FORMAT(51X,42HRMS VALUE / ENCOUNTERED MODAL PERIOD (TOE))
       IF (IPAGE .EQ. 2) GO TO 225

*    starboard headings

       WRITE (IPRIN,1042) (HEADNG(IH),IH=1,13)
 1042  FORMAT (/58X,29HSHIP HEADING ANGLE IN DEGREES/4X,1HV,2X,2HTO,7X,
     2 4HHEAD,47X,9HSTBD BEAM,46X,6HFOLLOW/10X,13(6X,I3))
       DO 220 IV=1,NVK
       IVK = VK(IV) + .5001
       WRITE (AVK,1045) IVK
 1045  FORMAT (I2)
       WRITE (IPRIN,1050)
 1050  FORMAT (1H )
       DO 220 ITO=1,NTMOD
       SWHMAX = .202*TMODAL(ITO)**2
       IF (PUNITS(1) .EQ. METER) SWHMAX = SWHMAX*FTMETR
       IF (SIGWH(IS) .GT. SWHMAX) GO TO 220
       IMP = TMODAL(ITO) + .5001
       DO 210 IH=1,13
       TEMRMS(IH) = RMSTBL(IH,ITO,IV)
       IF (IM.EQ.15) TEMRMS(IH) = TEMRMS(IH) * 100
       IF (IM.LT.4 .AND. IT.EQ.3)
     2 TEMRMS(IH) = TEMRMS(IH) * 100
       IF (IM .EQ. 7) TEMRMS(IH) = TEMRMS(IH) / 1000.
```

```
      IF (IP.GT.0 .AND. (IM.GE.10 .AND. IM.LE.11)) TEMRMS(IH) =
    2 TEMRMS(IH)/100
      IF (IP.GT.0 .AND. (IM.GE.12 .AND. IM.LE.14)) TEMRMS(IH) =
    2 TEMRMS(IH)/10000
      TEMTOE(IH) = TOETBL(IH,ITO,IV)
      IF(TEMTOE(IH) .GT. 99) TEMTOE(IH)=99
  210 CONTINUE
      WRITE (IPRIN,1052) AVK,IMP,(TEMRMS(IH),TEMTOE(IH),IH=1,13)
 1052 FORMAT (3X,A2,2X,I2,3X,13(1X,F5.2,1H/,I2))
      AVK = BLANK
  220 CONTINUE
      GO TO 250

*    port headings

  225 WRITE (IPRIN,1043) (HEADNG(IH),IH=14,26)
 1043 FORMAT (/58X,29HSHIP HEADING ANGLE IN DEGREES/4X,1HV,2X,2HTO,7X,
    2 4HHEAD,47X,9HPORT BEAM,46X,6HFOLLOW/10X,13(6X,I3))
      DO 240 IV=1,NVK
      IVK = VK(IV) + .5001
      WRITE (AVK,1045) IVK
      WRITE (IPRIN,1050)
      DO 240 ITO=1,NTMOD
      SWHMAX = .202*TMODAL(ITO)**2
      IF (PUNITS(1) .EQ. METER) SWHMAX = SWHMAX*FTMETR
      IF (SIGWH(IS) .GT. SWHMAX) GO TO 240
      IMP = TMODAL(ITO) + .5001
      LH = 26
      DO 230 IH=1,13
      LH = LH - 1
      TEMRMS(IH) = RMSTBL(LH,ITO,IV)
      IF (IM.EQ.15) TEMRMS(IH) = TEMRMS(IH) * 100
      IF ((IM.LT.4 .OR. IM.EQ.9) .AND. IT.EQ.3)
    2 TEMRMS(IH) = TEMRMS(IH) * 100
      IF (IM .EQ. 7) TEMRMS(IH) = TEMRMS(IH) / 1000.
      IF (IP.GT.0 .AND. (IM.GE.10 .AND. IM.LE.11)) TEMRMS(IH) =
    2 TEMRMS(IH)/100
      IF (IP.GT.0 .AND. (IM.GE.12 .AND. IM.LE.14)) TEMRMS(IH) =
    2 TEMRMS(IH)/10000
      TEMTOE(IH) = TOETBL(LH,ITO,IV)
      IF(TEMTOE(IH) .GT. 99) TEMTOE(IH)=99
  230 CONTINUE
      WRITE (IPRIN,1052) AVK,IMP,(TEMRMS(IH),TEMTOE(IH),IH=1,13)
      AVK = BLANK
  240 CONTINUE
  250 CONTINUE
  300 CONTINUE

*    change for VAX/VMS version
*   CDC     IF (JR .EQ. 0) GO TO 310
*   CDC     CALL STINDX (SEVFIL,RSINDX,NRSIND)
*   CDC     CALL WRITMS (SEVFIL,SWINDX,NSWIND,JR)
*   CDC310 CONTINUE

      IF (IM.EQ.8 .AND. IT.EQ.2) CALL DKWSLM (KR,IO,IM,NPREDH,N,NDATA,
    2 DATA,INDXRL,INDXHD,HEADNG,HDNG,LINEAR,SYMMET,SPINDX,TOINDX,IP,
    2 RMSTBL,TOETBL,RMS,ROLL)

  400 CONTINUE

*    change for VAX/VMS version
*   CDC     IF (ISKPSV .EQ. 1) GO TO 410
*   CDC     CALL STINDX (SEVFIL,SVIDX,LSVIDX)
*   CDC     CALL WRITMS (SEVFIL,RSINDX,NRSIND,IO)
*   CDC410 CONTINUE

  500 CONTINUE

      CLOSE (UNIT=RMSFIL)
      IF (ISKPSV .EQ. 0) CLOSE (UNIT=SEVFIL)
      CLOSE (UNIT=SPDFIL)
      CLOSE (UNIT=SPTFIL)
      IF (ISKPSV .EQ. 0) CALL SEVMOT (NSVRSP,RSPNME,HDNG,IMODL)

      RETURN
      END

C DECK RMSTOE
      SUBROUTINE RMSTOE

*    The purpose of the rmstoe segment is to compute the rms, second and
*    fourth moments, encounter spectra and associated periods of maximum
*    spectral energy for any ship response.  The calculations are done
*    for unit significant wave height in long and shotcrested seas for
*    a series of modal wave periods.  The shortcrested calculations are
*    performed using a cosine-squared weighting function.
*              W.G.MEYERS, DTNSRDC, 100777

      COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,LAOPR,RLDMPR,DISPLMT,
     2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
     2 NLEWF(25),NLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
     2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
     2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
     2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
     2 STATNM,STATIS
      CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
      INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
     2 FBNUMB,PTNUMB,ORGOPTN
      REAL KG

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
      INTEGER LPFIDX,LRMIDX,LSVIDX
      REAL PFIDX(230),RMIDX(183),SVIDX(3)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR,PUNITS(2)

      COMMON /RESPN/ NRESP,IPOINT(182),IMOTN(182),ITYPE(182),
     2 ILIN(182),ISYM(182)
      LOGICAL ILIN,ISYM

      COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
     2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
     2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
     2 LSHIPS,LTITLES
       CHARACTER*160 AS
       CHARACTER*30 FIS,SIS,SOS,SDS,TITLES
       CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
       CHARACTER SHIPS*6,VARS*2,CYCLS*2
       INTEGER*2 OPTION

       COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
       LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

       DIMENSION WEVN(100),SPINDX(9),TOINDX(9),DATA(432),AOMGE(30,13),
     2 R(30),RAO1(30,8,13),RAO2(30,8,11),KREC(13),B2(35)
       INTEGER DELBET
       LOGICAL LINEAR,SYMMET
       DIMENSION XID(911)
       EQUIVALENCE (NRESP,XID)

       NID = 911
       NWEVN = 100
       CALL WEDEFN (NWEVN,WEVN)
       DELBET = 15
       NLCH = 11
       CALL SCB2 (DELBET,B2,PI,NLCH)
       NPLANE = 2
       NSPIND = NVK + 1
       NTOIND = NTMOD + 1

       FIS = SDS(1:LSDS)//'.RMS'
       OPEN (UNIT=RMSFIL,FILE=FIS,STATUS='UNKNOWN',
     2 ACCESS='DIRECT',RECL=1750)

       FIS = SDS(1:LSDS)//'.ORG'
       OPEN (UNIT=ORGFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

       FIS = SDS(1:LSDS)//'.LCO'
       IF (LOADS)
     2 OPEN (UNIT=LCOFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

*          modified to run on VAX/VMS
*   CDC    CALL WRITMS (RMSFIL,XID,NID,1)

       WRITE (RMSFIL,REC=1) (XID(I),I=1,432)
       WRITE (RMSFIL,REC=2) (XID(I),I=433,796),NID
       WRITE (RMSFIL,REC=3) (XID(I),I=797,911)
       NRECD = 3

       DO 60 IR=1,NRESP
       LINEAR = ILIN(IR)
       SYMMET = ISYM(IR)
       NPREDH = 13
       IF (.NOT. SYMMET) NPREDH = 24
       JA = 1
       IF (.NOT. LINEAR) JA = 5
       N = 1
       IF (.NOT. LINEAR) N = NRANG
       IF (LOADS) REWIND LCOFIL
       REWIND ORGFIL
       READ (ORGFIL) TITLE,NVK,NMU,NOMEGA,OMEGA,NRANG,RLANG,VRT,LAT,
     2 ADDRES,LPP,BEAM,DRAFT,DISPLM,GM,DELGM,KG,KROLL,LCB,GRAV,RHO,
     2 VKDES,VKINC,DBLWL

*     define 2-parameter (significant wave height, modal wave period)
*     Bretschneider sea spectra, for unit significant wave height

       DO 500 IT=1,NTMOD
       CALL BRWVSP (NOMEGA,1.,TMODAL(IT),OMEGA,S(1,IT))
  500  CONTINUE

       IPHS = 0
       NDATA = (2 + N*2)*NPREDH

*          modified for VAX/VMS
*  CDC      CALL STINDX (RMSFIL,SPINDX,NSPIND)
*  CDC      DO 10 I=1,NSPIND
*  CDC      SPINDX(I) = 0.
*  CDC 10   CONTINUE

        DO 60 IV=1,NVK
        NMU = NNMU(IV)
        NLCH = NMU - 2
        N1 = NMU/2 - 1
        N2 = NMU/2 + 1
        NBETA = 2*(NMU-1)

        DO 15 IH=1,NMU
        KH = IH - 1
        IF (IH .EQ. 1) KH = 1
        IF (IH .EQ. 13) KH = 11
        CALL RAOPHS (AOMGE(1,IH),RAO1(1,1,IH),DUM,RAO2(1,1,KH),DUM,
      2 KREC(IH),IR,IV,IH,IPHS)
   15   CONTINUE

*          modified for VAX/VMS
*  CDC      CALL STINDX (RMSFIL,TOINDX,NTOIND)
*  CDC      DO 20 I=1,NTOIND
*  CDC      TOINDX(I) = 0.
*  CDC 20   CONTINUE

        DO 40 IT=1,NTMOD

        CALL RMS (KREC,RAO1,RAO2,IT,N,R,B2,NPREDH,NLCH,N1,N2,DATA,
      2 IMOTN(IR),NBETA)

        CALL TOE (KREC,AOMGE,RAO1,RAO2,JA,IT,R,B2,NPREDH,
      2 NLCH,N1,N2,NBETA,DELBET,NWEVN,WEVN,IV,DATA)

*          modified for VAX/VMS
*  CDC      CALL WRITMS (RMSFIL,DATA,NDATA,IT)

        NRECD = NRECD + 1
        WRITE (RMSFIL,REC=NRECD) DATA
   40   CONTINUE

*  CDC      CALL STINDX (RMSFIL,SPINDX,NSPIND)
*  CDC      CALL WRITMS (RMSFIL,TOINDX,NTOIND,IV)

   50   CONTINUE

*  CDC      CALL STINDX (RMSFIL,RMIDX,LRMIDX)

        KR = IR + 1

*  CDC      CALL WRITMS (RMSFIL,SPINDX,NSPIND,KR)

   60   CONTINUE

        CLOSE (UNIT=RMSFIL)
        CLOSE (UNIT=ORGFIL)
        IF (LOADS) CLOSE (UNIT=LCOFIL)

        RETURN
        END

C DECK RPHI2D
        SUBROUTINE RPHI2D (K,PHI2D)

        COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTS1,
      2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
        REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
        INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

        COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
      1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
        INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
      INTEGER LPFIDX,LRMIDX,LSVIDX
      REAL PFIDX(235),RMIDX(183),SVIDX(3)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

      COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
      LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

      COMPLEX PHI2D(10,10,4)
      REAL DATA(320)

      NNODE = NOFSET(K)
      NDATP = 0
      IF (VRT) NDATP = 16*NNODE
      IF (LAT) NDATP = NDATP + 16*NNODE
      ISIGMX = NSIGMA - 1
      DO 30 ISIGMA=1,ISIGMX
      INDEX = (ISIGMA-1)*NSTATN + K

*          modified for VAX/VMS
*  CDC    CALL READMS (POTFIL,DATA,NDATP,INDEX)
      READ (POTFIL,REC=INDEX) DATA

      NEXT = 1
      DO 20 J=1,NNODE
      DO 10 I=IMMIN,IMMAX,IMDEL
      PHI2D(ISIGMA,J,I) = CMPLX(DATA(NEXT),DATA(NEXT+1))
      IF (ISIGMA .EQ. ISIGMX) PHI2D(NSIGMA,J,I) =
     2 CMPLX(DATA(NEXT+4),DATA(NEXT+5))
      NEXT = NEXT + 8
   10 CONTINUE
   20 CONTINUE
   30 CONTINUE

      RETURN
      END

C DECK RSOLVE
      SUBROUTINE RSOLVE( N, NDIM, A, B, IP )

*        solution of linear system, A*X = B .

*        INPUT...
*           N = order of matrix.
*           NDIM = declared dimension of array  A .
*           A = triangularized matrix obtained from "DECOMP".
*           B = right hand vector.
*           IP = POVOT vector obtained from "DECOMP".
*        do not use solve if DECOMP has set IP(N) = 0 .
*        OUTPUT...
*           B = solution vector, X .

       REAL A, B, T
       INTEGER N, NDIM, IP, I, K, KB, KM1, KP1, M, NM1
       DIMENSION A(NDIM,NDIM), B(NDIM)
       DIMENSION IP(NDIM)

       IF (N .EQ. 1) GO TO 1600
       NM1 = N - 1
       DO 1200 K = 1, NM1
       KP1 = K + 1
       M = IP(K)
       T = B(M)
       B(M) = B(K)
       B(K) = T
       DO 1100 I = KP1, N
       B(I) = B(I) + A(I,K)*T
1100   CONTINUE
1200   CONTINUE
       DO 1400 KB = 1, NM1
       KM1 = N - KB
       K = KM1 + 1
       B(K) = B(K)/A(K,K)
       T = -B(K)
       DO 1300 I = 1, KM1
       B(I) = B(I) + A(I,K)*T
1300   CONTINUE
1400   CONTINUE
1600   CONTINUE
       B(1) = B(1)/A(1,1)
99999  CONTINUE

       RETURN
       END

C DECK RSTITL
       SUBROUTINE RSTITL (IP,IM,IT,RTITL,RTYPE,RUNIT,PARS)

       COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
     2 LRAOPR,ADRPR,ORGOPTN,GMNCM,KG,STATN(25),NSOFST(25),
     2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
     2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
     2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
     2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
     2 STATNM,STATIS
       CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
       INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
     2 FBNUMB,PTNUMB,ORGOPTN
       REAL KG

       COMMON /LOADS/ NLOADS,SWGHT(25),SMASS(25),XLDSTN(10),XLDXPT(25),
     2 LSTATN(25)

       COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
       COMPLEX II
       CHARACTER*4 PUNITS(2)
       REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

       CHARACTER*3 PT(10),TT(3),PPM(3),RLT(2)
       CHARACTER*4 METER,MUNIT(3,7),PNTMOT(2,3),LOAD(2,5)
       CHARACTER*4 LTYPE(3,2),LUNIT(3,3),TYPE(3,5),RELMOT(2)
       CHARACTER*4 ADRES(2),ADRTYP(3),UNIT(3,7),RUNIT(3)
       CHARACTER*4 RTITL(2),RTYPE(3),HFEMOT(2)
       CHARACTER*5 PPLM(3),TTLM(3),HFEM,HHFEM
       CHARACTER*6 ORGMOT(2,6)
       CHARACTER*10 OMOT(3,6),FMOT(3)
       CHARACTER*110 PARS

       DATA METER /'METE'/
```

```
      DATA MUNIT   /'  (M','ETER','S)  ','(MET','ERS/','SEC)','    ',
     2 '(G) ','    ','   (','DEG)','    ',' (DE','G/SE','C)  ','(DEG',
     2 '/SEC','2)  ','   (','LBS)','    '/
      DATA PPLM /'LONG.','LATE.','VERT.'/
      DATA HFEM /'HOR'/
      DATA HHFEM /'HORZ.'/
      DATA TTLM /'DISP.','VEL. ','ACC. '/
      DATA PT /'P1','P2','P3','P4','P5','P6','P7','P8','P9','P10'/
      DATA TT /'DSP','VEL','ACC'/
      DATA PPM /'LON','LAT','VER'/
      DATA OMOT /'SURGE','SURVEL','SURACC','SWAY','SWAVEL','SWAACC',
     2 'HEAVE','HEAVEL','HEAACC','ROLL','ROLVEL','ROLACC','PITCH',
     2 'PITVEL','PITACC','YAW','YAWVEL','YAWACC'/
      DATA ORGMOT /'   S','URGE','    ','SWAY','   H','EAVE','    ',
     2 'ROLL','   P','ITCH','    ',' YAW'/
      DATA PNTMOT /' LON','GIT.',' LAT','ERAL','VERT','ICAL'/
      DATA HFEMOT /' HOR','IZ. '/
      DATA FMOT /'FINANG','FINVEL','FINACC'/
      DATA LOAD /' H.S','HEAR',' V.S','EEAR','   T','ORS.',' V.B',
     2 'END.',' H.B','END.'/
      DATA LTYPE /'FORC','E   ','    ','MOME','NT  ','    '/
      DATA LUNIT /'  (T','ONS)','    ',' (M-','TONS',')   ',' (FT',
     2 '-TON','S)  '/
      DATA TYPE /'DISP','LACE','MENT','VELO','CITY','    ','ACCE',
     2 'LERA','TION','ANGL','E   ','    ','MOTI','ON  ','    '/
      DATA RLT /'RLM','RLV'/
      DATA RELMOT /'RELA','TIVE'/
      DATA ADRES /'   A','DDED'/
      DATA ADRTYP /'RESI','STAN','CE  '/
      DATA UNIT /'  (F','EET)','    ','(FEE','T/SE','C)  ','    ',
     2 '(G) ','    ','   (','DEG)','    ',' (DE','G/SE','C)  ','(DEG',
     2 '/SEC','2)  ','   (','LBS)','    '/

      RUNIT(1) = UNIT(1,IT)
      RUNIT(2) = UNIT(2,IT)
      RUNIT(3) = UNIT(3,IT)
      IF (PUNITS(1) .EQ. METER) RUNIT(1) = MUNIT(1,IT)
      IF (PUNITS(1) .EQ. METER) RUNIT(2) = MUNIT(2,IT)
      IF (PUNITS(1) .EQ. METER) RUNIT(3) = MUNIT(3,IT)
      JT = IT + 3
      IF (IP .GT. 0) GO TO 20

      IF (IM .GT. 6) GO TO 10

*     origin motions

      RTITL(1) = ORGMOT(1,IM)
      RTITL(2) = ORGMOT(2,IM)
      RTYPE(1) = TYPE(1,IT)
      RTYPE(2) = TYPE(2,IT)
      RTYPE(3) = TYPE(3,IT)
      IF (IM.GT.3 .AND. IT.EQ.1) RTYPE(1) = TYPE(1,4)
      IF (IM.GT.3 .AND. IT.EQ.1) RTYPE(2) = TYPE(2,4)
      IF (IM.GT.3 .AND. IT.EQ.1) RTYPE(3) = TYPE(3,4)
      IF (IM .GT. 3) RUNIT(1) = UNIT(1,JT)
      IF (IM .GT. 3) RUNIT(2) = UNIT(2,JT)
      IF (IM .GT. 3) RUNIT(3) = UNIT(3,JT)
      IF (PUNITS(1) .EQ. METER .AND. IM .GT. 3) RUNIT(1) = MUNIT(1,JT)
      IF (PUNITS(1) .EQ. METER .AND. IM .GT. 3) RUNIT(2) = MUNIT(2,JT)
      IF (PUNITS(1) .EQ. METER .AND. IM .GT. 3) RUNIT(3) = MUNIT(3,JT)
      WRITE (PARS,3000) (OMOT(IT,IM),J=1,2)
 3000 FORMAT (A10,10X,A10,80X)
      GO TO 60

   10 IF (IM .NE. 7) GO TO 30

*     added resistance

      RTITL(1) = ADRES(1)
      RTITL(2) = ADRES(2)
      RTYPE(1) = ADRTYP(1)
      RTYPE(2) = ADRTYP(2)
```

```
        RTYPE(3) = ADRTYP(3)
        RUNIT(1) = UNIT(1,7)
        RUNIT(2) = UNIT(2,7)
        RUNIT(3) = UNIT(3,7)
        GO TO 50

   20   IF (IM .GT. 3) GO TO 30

*     motions at a point

        RTITL(1) = PNTMOT(1,IM)
        RTITL(2) = PNTMOT(2,IM)
        RTYPE(1) = TYPE(1,IT)
        RTYPE(2) = TYPE(2,IT)
        RTYPE(3) = TYPE(3,IT)
        WRITE (PARS,3010) PPM(IM),TT(IT),PT(IP),PPLM(IM),TTLM(IT),
     2 XPTLOC(IP),YPTLOC(IP),ZPTLOC(IP)
 3010   FORMAT (3A3,11X,A5,1X,A5,4X,2HAT,4X,5HXFP =,F6.2,3X,5HYCL =,
     2 F7.2,3X,5HZBL =,F7.2,28X)
        GO TO 50

   30   IF (IM .NE. 8) GO TO 50

*     relative motion

        RTITL(1) = RELMOT(1)
        RTITL(2) = RELMOT(2)
        RTYPE(1) = TYPE(1,IT)
        RTYPE(2) = TYPE(2,IT)
        RTYPE(3) = TYPE(3,IT)
        IF (IT .EQ. 1) RTYPE(1) = TYPE(1,5)
        IF (IT .EQ. 1) RTYPE(2) = TYPE(2,5)
        IF (IT .EQ. 1) RTYPE(3) = TYPE(3,5)
        WRITE (PARS,3020) RLT(IT),PT(IP),RTITL,RTYPE,XPTFBD(IP),
     2 YPTFBD(IP),ZPTFBD(IP)
 3020   FORMAT (2A3,14X,2A4,1X,3A4,2HAT,4X,5HXFP =,F6.2,3X,5HYCL =,
     2 F7.2,3X,5HZBL =,F7.2,22X)

   50   IF (IM .NE. 9) GO TO 72

*     anti-roll fins

        RTITL(1) = '    '
        RTITL(2) = ' FIN'
        IF (IT .EQ. 1) JT = 4
        IF (IT .GT. 1) JT = IT
        DO 60 I=1,3
   60   RTYPE(I) = TYPE(I,JT)
        JT = IT + 3
        DO 70 I=1,3
   70   RUNIT(I) = UNIT(I,JT)
        WRITE (PARS,3000) FMOT(IT),FMOT(IT)

   72   IF (IM .NE. 15) GO TO 80
        RTITL(1) = HFEMOT(1)
        RTITL(2) = HFEMOT(2)
        RTYPE(1) = TYPE(1,3)
        RTYPE(2) = TYPE(2,3)
        RTYPE(3) = TYPE(3,3)
        RUNIT(1) = UNIT(1,3)
        RUNIT(2) = UNIT(2,3)
        RUNIT(3) = UNIT(3,3)
        WRITE (PARS,3010) HFEM,TT(3),PT(IP),HHFEM,TTLM(3),
     2 XPTLOC(IP),YPTLOC(IP),ZPTLOC(IP)

   80   IF (.NOT. (IP.GT.0.AND.(IM.GE.10.AND.IM.LE.14))) GO TO 100

*     loads

        JM = IM - 9
        RTITL(1) = LOAD(1,JM)
        RTITL(2) = LOAD(2,JM)
```

```
      LT = 1
      IF (IM .GT. 11) LT = 2
      MT = LT
      IF (LT.EQ.2.AND.(PUNITS(1).NE.METER)) MT = 3
      DO 82 I=1,3
      RTYPE(I) = LTYPE(I,LT)
      RUNIT(I) = LUNIT(I,MT)
   82 CONTINUE
      IF (JM .EQ. 1) WRITE (PARS,3031) PT(IP),XLDSTN(IP)
      IF (JM .EQ. 2) WRITE (PARS,3032) PT(IP),XLDSTN(IP)
      IF (JM .EQ. 3) WRITE (PARS,3033) PT(IP),XLDSTN(IP)
      IF (JM .EQ. 4) WRITE (PARS,3034) PT(IP),XLDSTN(IP)
      IF (JM .EQ. 5) WRITE (PARS,3035) PT(IP),XLDSTN(IP)
 3031 FORMAT(6HHSHEAR,A3,11X,29HHORIZ. SHEAR FORCE AT STATION,F6.2,55X)
 3032 FORMAT(6HVSHEAR,A3,11X,29HVERT.  SHEAR FORCE AT STATION,F6.2,55X)
 3033 FORMAT(4HTMOM,  A3,13X,29HTORSIONAL MOMENT   AT STATION,F6.2,55X)
 3034 FORMAT(4HVMOM,  A3,13X,29HVERT.  BEND. MOM.  AT STATION,F6.2,55X)
 3035 FORMAT(4HHMOM,  A3,13X,29HHORIZ. BEND. MOM.  AT STATION,F6.2,55X)
  100 CONTINUE

      RETURN
      END

C DECK RVSLAT
      SUBROUTINE RVSLAT (VCG,MOTLG,MOTL)

      COMPLEX MOTLG(3),MOTL(3)

      MOTL(1) = MOTLG(1) + VCG*MOTLG(2)
      MOTL(2) = MOTLG(2)
      MOTL(3) = MOTLG(3)

      RETURN
      END


C DECK SBEDDY
      SUBROUTINE SBEDDY

      COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
     2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
     2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
     2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
     2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
     2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2)
     2,SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
     2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
     2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
     2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
     2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

      COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
     2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
     2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
     2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
     2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
     2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
     2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
     2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
     2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
     2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
     2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
     2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
     2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
     2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
     2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
     2 ENSHF(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
       REAL RDBLK(2692)
       EQUIVALENCE (PSUR(1),RDBLK(1))

       DO 20 IA=1,NRANG
       ENPE(IA) = 0
       DO 10 IS=1,NSIGMA
       SHPDMP(IS,IA) = 0
   10  CONTINUE
   20  CONTINUE
       IF (NSBSET .EQ. 0) GO TO 100
       DO 60 K=1,NSBSET
       DO 50 L=1,2
       IF (L.EQ.2 .AND. SBTHB(K).EQ.0.) GO TO 50
       YHAT = SQRT(PYCP(K,L)**2 + PZCP(K,L)**2)
       GAMMAE = PGAMMA(K,L) + 1.
       ALF = ATAN( ABS( ((PYCP(K,L)/PZCP(K,L)) + TAN(GAMMAE*DEGRAD)) /
     2 (1. - (PYCP(K,L)/PZCP(K,L))*TAN(GAMMAE*DEGRAD)) ) )
       C = 0.0065 + (PLCS(K,L)**2)/(0.9*PI*PEAR(K,L))
       CON = PQ(K,L)*4./(3.*PI)*RHO*YHAT**3 * PAREA(K,L)*C*SIN(ALF)
       DO 40 IA=1,NRANG
       DO 30 IS=1,NSIGMA
       SHPDMP(IS,IA) = SHPDMP(IS,IA) + (CON*SIGMA(IS)*RANG(IA)) *
     2 SIGMA(IS)
   30  CONTINUE
   40  CONTINUE
   50  CONTINUE
   60  CONTINUE
       DO 70 IA=1,NRANG
       CALL SPFIT (SIGMA,SHPDMP(1,IA),PEELM(1,1,IA),NSIGMA)
       ENPE(IA) = ENCON*REVAL(PEELM(1,ISIGMA,IA),WTSI)
   70  CONTINUE
  100  CONTINUE

       RETURN
       END

C DECK SBLIFT
       SUBROUTINE SBLIFT

       COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
     2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
     2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
     2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
     2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
     2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2)
     2,SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
     2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
     2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
     2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
     2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

       COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
       INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
       REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

       COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
       INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
       CHARACTER*4 TITLE(20)
       REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

       COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
       COMPLEX II
       CHARACTER*4 PUNITS(2)
       REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

       COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
     2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
     2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
     2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
     2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
     2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
     2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
     2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
     2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
     2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
     2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
     2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
     2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
     2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
     2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
       REAL RDBLK(2692)
       EQUIVALENCE (PSUR(1),RDBLK(1))

       REAL LCS,MCHORD

       IF (NSBSET .EQ. 0) GO TO 60
       EN = 0
       STASPC = LPP/20
       DO 50 K=1,NSBSET
       DO 40 L=1,2
       IF (L.EQ.2 .AND. SBTHB(K).EQ.0.) GO TO 40
       IF (L .EQ. 2) GO TO 20

*     outer brackets

       XRTF = LCB - SOBRFS(K)*STASPC
       XRTA = LCB - SOBRAS(K)*STASPC
       XTPF = LCB - SBTFS(K)*STASPC
       XTPA = LCB - SBTAS(K)*STASPC
       YRT = SOBRHB(K)
       YTP = SBTHB(K)
       ZRT = (SOBRFW(K) + SOBRAW(K))/2 - (DBLWL+VCG)
       ZTP = (SBTFWL(K) + SBTAWL(K))/2 - (DBLWL+VCG)
       GO TO 30

*     inner bracket

   20  XRTF = LCB - SIBRFS(K)*STASPC
       XRTA = LCB - SIBRAS(K)*STASPC
       YRT = SIBRHB(K)
       ZRT = (SIBRFW(K) + SIBRAW(K))/2 - (DBLWL+VCG)
   30  CONTINUE
       RCHORD = XRTF - XRTA
       TCHORD = XTPF - XTPA
       SPAN = SQRT((ZRT-ZTP)**2 + (YTP-YRT)**2)
       Q = 2
       MCHORD = 0.5*((XRTF-XRTA) + (XTPF-XTPA))

*     area
```

```
      AREA = SPAN*MCHORD

*     center of pressure

      ZP = 0.5*(ZRT+ZTP)
      YP = 0.5*(YRT + YTP)
      XO = 0.5*(XRTF + XTPF)
      XCP = XO - 0.25*MCHORD
      YCP = YP
      ZCP = ZP

*     moment arm

      ARG = (ZRT-ZTP) / SPAN
      GAMMA = - 90
      IF (ARG .LT. 1) GAMMA = - ASIN(ARG)*RADDEG
      IF (L .EQ. 1) GAMMA = - GAMMA
      GAM = GAMMA*DEGRAD
      YHAT = YCP*COS(GAM) + ZCP*SIN(GAM)

*     effective aspect ratio

      EAR = 2*SPAN/MCHORD

*     lift curve slope

      LCS = 2*PI
      PQ(K,L) = Q
      PSPAN(K,L) = SPAN
      PMNCHD(K,L) = MCHORD
      PAREA(K,L) = AREA
      PXCP(K,L) = XCP
      PYCP(K,L) = YCP
      PZCP(K,L) = ZCP
      PGAMMA(K,L) = GAMMA
      PYHAT(K,L) = YHAT
      PEAR(K,L) = EAR
      PLCS(K,L) = LCS
      EN = EN + Q*(RHO/2)*AREA*LCS*YHAT*YHAT*WPHI*ENCON
  40  CONTINUE
  50  CONTINUE
  60  CONTINUE
      DO 70 IV=1,NVK
      ENPL(IV) = 0
      IF (NSBSET .GT. 0) ENPL(IV) = EN*VFS(IV)
  70  CONTINUE

      RETURN
      END

C DECK SCB2
      SUBROUTINE SCB2 (DELHDG,B2,PI,NLCH)

*     This routine pre-computes the shortcrested weighting
*     constants, B2, for variable spreading angles.
*            W.G.MEYERS, DTNSRDC, 072977

      INTEGER DELHDG
      DIMENSION B2(NLCH)

      N = 180/(2*DELHDG)
      CON1 = 1./N
      CON2 = PI/(2*N)
      I = - N
      DO 10 K=1,NLCH
      I = I + 1
      COSI = COS(I*CON2)
      B2(K) = CON1*COSI*COSI
  10  CONTINUE

      RETURN
      END
```

```
C DECK SECT1
      SUBROUTINE SECT1

*  determines section type (ITSK) and bilge radius (RDK)
*     ITSK = 1  bow sections - narrow v or u
*     ITSK = 2  full sections
*     ITSK = 3  shallow v or u  (destroyer stern)
*     ITSK = 4  very rounded destroyer midship section - no eddymaking

      COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
     2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
     2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
     2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
     2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
     2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2)
     2,SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
     2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
     2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
     2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
     2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

      COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
     2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
     2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
     2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
     2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
     2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
     2 STATNM,STATIS
      CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
      INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
     2 FBNUMB,PTNUMB,ORGOPTN
      REAL KG

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,CML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

      COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
     2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
     2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
     2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
     2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
     2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
     2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
     2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
     2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
     2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
     2 STADMP(10),SHPDMP(10,8),ENCON,WTTI,TPHI,WMELM(4,9),SFELM(4,9,8),
     2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
     2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
     2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
     2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
      REAL RDBLK(2692)
      EQUIVALENCE (PSUR(1),RDBLK(1))
      DIMENSION AA(3,4),AR(10)

      M = NSTATN + 1
      DO 100  K=1,NSTATN
      M = M - 1
      ITSK=4
      RDK=1.
      IF (NOFSET(K) .LT. 2)  GO TO 21
      NNODES = NOFSET(K)
      BLOCAL = BMK(K)
      TLOCAL = DK(K)
      ORG = TLOCAL - VCG
      CAC = CAK(K)
      GDB = ABS(ORG)/(2.*BLOCAL)
      RMIN=1.E38
      NNM=NNODES-1
      DO 31  I=2,NNM
      DO 32  J=1,3
      IDX=I+J-2
      AA(J,1)=Y(IDX,K)**2+Z(IDX,K)**2
      AA(J,2)=Y(IDX,K)
      AA(J,3)=Z(IDX,K)
      AA(J,4)=1.0
  32  CONTINUE
      A=CMINR(1,AA)
      B=-CMINR(2,AA)
      C=CMINR(3,AA)
      D=-CMINR(4,AA)
      IF (A .EQ. 0)  GO TO 33
      DY=Y(I+1,K)-Y(I-1,K)
      IF (ABS(DY) .EQ. 0.) GO TO 33
      ZT=Z(I-1,K)+(Z(I+1,K)-Z(I-1,K))*(Y(I,K)-Y(I-1,K))/DY
      IF(ZT.LE.Z(I,K))  GO TO 33
      YC=-B/(2.*A)
      ZC=-C/(2.*A)
      R=SQRT(ABS(YC*YC+ZC*ZC-D/A))
      AR(I) = R
      IF (R .LT. RMIN)  RMIN=R
  33  CONTINUE
  31  CONTINUE
      RDK=RMIN

*   SERE not used (triangular sections)
*     IF (BDG.GT.0.8 .AND. BDG.LE.2.25) ITSK = 3

      IF (CAC .GT. 0.65) ITSK = 4
      IF (CAC .GE. 0.95) ITSK = 2
      IF (GDB .GE. 1.2) ITSK = 1

*   no eddymaking (TANAKA) for stations with bilgekeels

      IF (NBKSET .EQ. 0) GO TO 40
      DO 30 I=1,NBKSET
      NBKS = NBKSTN(I)
      DO 20 J=1,NBKS
      IF (.NOT.(STATN(M).EQ.BKSTN(J,I))) GO TO 20
      YBK = BKHB(J,I)
      ZBK = BKWL(J,I) - DBLWL

      WRITE (IPRIN,1000) BKSTN(J,I),YBK,ZBK
 1000 FORMAT (/3F10.2)

      M1=2
      M2=NNM
      DO 11 NN=2,NNM
      IF (Z(NN,K).LT.ZBK) GO TO 11
      M2=NN
      M1=NN-1
      IF (Z(NN,K).EQ.ZBK) M2=NN+1
      GO TO 12
  11  CONTINUE
  12  CONTINUE
```

```
      L = NNODE
      DO 13 NN=2,NNM
      L = L - 1
      R = AR(L)
      WRITE (IPRIN;1010) Y(L,K),Z(L,K),AR(L)
1010  FORMAT (2F10.2,1PE12.2)
  13  CONTINUE

      WRITE(IPRIN,1011) M1,M2
1011  FORMAT ('  M1, M2 = ',2I5)

*    search for minimum radius of the bilge starting from the waterline

      RMIN = AR(M2)
      L = M2+1
      DO 15 NN=M1,M2
      L = L - 1
      R = AR(L)
      IF (R .GT. RMIN) GO TO 17
      RMIN = R
  15  CONTINUE
  17  RDK = RMIN

      WRITE (IPRIN,1020) RMIN
1020  FORMAT (8H  RMIN =,1PE12.2)

      ITSK = 4
      GO TO 21
  20  CONTINUE
  30  CONTINUE
  40  CONTINUE

*    SERE used for sections with skegs

      IF (NSKSET .EQ. 0) GO TO 60
      DO 50 I=1,NSKSET
      IF (STATN(M) .LE. SKAUS(I) .AND. STATN(M) .GE. SKFLS(I)) ITSK = 5
  50  CONTINUE
  60  CONTINUE
  21  CONTINUE
      RD(K)=RDK
      ITS(K)=ITSK
 100  CONTINUE

      RETURN
      END

C DECK SERAB
      SUBROUTINE SERAB (K,ROLANG,BLOCAL,TLOCAL,ORG,RD,EDDY,RGB)

      EXTERNAL EXP

*  calculates eddy-making roll damping data for TANAKA series A and B
*  REF- TANAKA, J. ZOSEN KIOKAI, VOL. 109, 1961

      RGB = SQRT(ORG*ORG + BLOCAL*BLOCAL) - RD*(SQRT(2.)-1.)
      BDG = 2.*BLOCAL/ABS(ORG)
      C = FIG56(ROLANG,BDG)*EXP(-FIG7(ROLANG)*RD/ABS(TLOCAL))
      C = C*FTWO(K,TLOCAL,RD)
      EDDY = C

      RETURN
      END

C DECK SERD
      SUBROUTINE SERD (K,ROLANG,BLOCAL,TLOCAL,ORG,EDDY,RGB)

      EXTERNAL EXP

*  calculates eddy-making roll damping data for TANAKA series D
*  REF- TANAKA, J. ZOSEN KIOKAI, VOL. 109, 1961
```

```
      RGB = ABS(ORG)
      IF (BLOCAL .LE. 0.) C = 0.63
      IF (BLOCAL .LE. 0.) GO TO 10
      GDB = RGB/(2.*BLOCAL)
      REQ = FIG10(GDB)*BLOCAL
      BDG = 1./GDB
      C = FIG56(ROLANG,BDG)*EXP(-FIG7(ROLANG)*REQ/ABS(TLOCAL))
   10 CONTINUE
      C = C*FTWO(K,TLOCAL,REQ)
      EDDY = C

      RETURN
      END

C DECK SERE
      SUBROUTINE SERE (BLOCAL,ORG,EDDY,RGB)

*  calculates oddy-making roll damping data for TANAKA series E
*  REF- TANAKA, J. ZOSEN KIOKAI, VOL. 109, 1961

      RGB = ABS(ORG)
      BDG = 2.*BLOCAL/ABS(ORG)
      C = FIG11(BDG)
      EDDY = C

      RETURN
      END

C DECK SETSEV
      SUBROUTINE SETSEV (NSVRSP,LSVRSP)

      COMMON /RESPN/ NRESP,IPOINT(182),IMOTN(182),ITYPE(182),
     2 ILIN(182),ISYM(182)
      LOGICAL ILIN,ISYM

      DIMENSION LSVRSP(NSVRSP)

      DO 160 LR=1,NSVRSP
      DO 140 IR=1,NRESP
      IP = IPOINT(IR)
      IM = IMOTN(IR)
      IT = ITYPE(IR)
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130),LR

10    IF (.NOT. (IP.EQ.0 .AND. IM.EQ.3 .AND. IT.EQ.1)) GO TO 140

*    heave
      GO TO 150

20    IF (.NOT. (IP.EQ.0 .AND. IM.EQ.5 .AND. IT.EQ.1)) GO TO 140

*    pitch
      GO TO 150

30    IF (.NOT. (IP.EQ.0 .AND. IM.EQ.2 .AND. IT.EQ.1)) GO TO 140

*    sway
      GO TO 150

40    IF (.NOT. (IP.EQ.0 .AND. IM.EQ.4 .AND  IT.EQ.1))  GO TO 140

*    roll
      GO TO 150

50    IF (.NOT. (IP.EQ.0 .AND. IM.EQ.6 .AND. IT.EQ.1))  GO TO 140

*    yaw
      GO TO 150

60    IF (.NOT. (IP.EQ.1 .AND. IM.EQ.3 .AND. IT.EQ.3))  GO TO 140
```

```
*     vertical acceleration at point 1 (p1)
        GO TO 150

70      IF (.NOT. (IP.EQ.1 .AND. IM.EQ.2 .AND. IT.EQ.3)) GO TO 140

*     lateral acceleration at point 1 (p1)
        GO TO 150

80      IF (.NOT. (IP.EQ.2 .AND. IM.EQ.3 .AND. IT.EQ.3))  GO TO 140

*     vertical acceleration at point 2 (p2)
        GO TO 150

90      IF (.NOT. (IP.EQ.2 .AND. IM.EQ.2 .AND. IT.EQ.3)) GO TO 140

*     lateral acceleration at point 2 (p2)
        GO TO 150

100     IF (.NOT. (IP.EQ.3 .AND. IM.EQ.3 .AND. IT.EQ.3)) GO TO 140

*     vertical acceleration at point 3 (p3)
        GO TO 150

110     IF (.NOT. (IP.EQ.3 .AND. IM.EQ.2 .AND. IT.EQ.3))  GO TO 140

*     lateral acceleration at point 3 (p3)
        GO TO 150

120     IF (.NOT. (IP.EQ.4 .AND. IM.EQ.3 .AND. IT.EQ.3)) GO TO 140

*     vertical acceleration at point 4 (p4)
        GO TO 150

130     IF (.NOT. (IP.EQ.4 .AND. IM.EQ.2 .AND. IT.EQ.3)) GO TO 140

*     lateral acceleration at point 4 (p4)
        GO TO 150

140     CONTINUE
150     LSVRSP(LR) = IR
160     CONTINUE

        RETURN
        END

C DECK SEVMOT
        SUBROUTINE SEVMOT (NSVRSP,RSPNME,HDNG,IMODL)

        COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
     2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSCFST(25),
     2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
     2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
     2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
     2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
     2 STATNM,STATIS
        CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
        INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
     2 FBNUMB,PTNUMB,ORGOPTN
        REAL KG

        COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
        INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
        REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

        COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
        INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
```

```
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
    2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
    2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
    4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
    5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
      INTEGER LPFIDX,LRMIDX,LSVIDX
      REAL PFIDX(235),RMIDX(183),SVIDX(3)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
    2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
    2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
    2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
    2 SPTFIL,LACFIL,LAEFIL

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
    2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
    1 RHOF,GNUS,GNUF,FTMETR

      COMMON /SEVERE/ NRSIND,RSINDX,NSWIND,SWINDX,RSVTOE,RV,RH
      REAL RSINDX(14),SWINDX(5),RSVTOE(402)
      INTEGER RV(13),RH(13)

      COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
    2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
    2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
    2 LSHIPS,LTITLES
      CHARACTER*160 AS
      CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
      CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
      CHARACTER SHIPS*6,VARS*2,CYCLS*2
      INTEGER*2 OPTION

      DIMENSION RSV(13,13),TOE(13,13),TEMV(13),TEMH(13),TEMR(13),
    2 TEMT(13),LSVRSP(13),HDNG(24),IMODL(4)
      CHARACTER*4 RSPNME(2,13)
      INTEGER TEMT
      CHARACTER*4 METER

      DATA METER /'METE'/
      DATA LSVRSP /2,4,1,3,5,7,6,9,8,11,10,13,12/

      FIS = SDS(1:LSDS)//'.SEV'
      OPEN (UNIT=SEVFIL,FILE=FIS,STATUS='UNKNOWN',
    2 ACCESS='DIRECT',RECL=1620)

      NHEAD = 24
      N1 = NHEAD + 1
      NDATA = 2 + N1*NVK*2
      DO 500 IC=1,2
      DO 400 IS=1,NSIGWH
      LT = IMODL(IS)
      DO 300 IR=1,NSVRSP
      DO 200 JR=1,NSVRSP
      LR = LSVRSP(JR)
      INDEX = NSIGWH * NSVRSP * (IC - 1) + NSIGWH * (LR - 1) + IS
      READ (SEVFIL,REC=INDEX) RSVTOE

*  CDC    CALL FETCH (IC,LR,IS,RSVTOE,SVIDX,RSINDX,SWINDX,NDATA,LSVIDX,
*  CDC  2 NSVRSP,NSIGWH,SEVFIL)

      IF (IR .GT. 1) GO TO 10
      RV(JR) = RSVTOE(1) + .001
      RH(JR) = RSVTOE(2)+ .001
   10 IF (JR .GT. 1) GO TO 20
      IV = RV(IR)
      IH = RH(IR)
  20  IE = 3 + (IH-1)*2 + (IV-1)*NHEAD*2
      RSV(JR,IR) = RSVTOE(IE)
      TOE(JR,IR) = RSVTOE(IE+1)
 200  CONTINUE
 300  CONTINUE
      WRITE (IPRIN,1000) TITLE
1000  FORMAT (1H1,/,28X,20A4,///,48X,28HS E V E R E   M O T I O N   ,
    2 9HT A B L E)
      IF (IC .EQ. 1) WRITE (IPRIN,1010)
      IF (IC .EQ. 2) WRITE (IPRIN,1020)
1010  FORMAT (//,60X,11HLONGCRESTED)
1020  FORMAT (//,60X,12HSHORTCRESTED)
      IF (PUNITS(1) .NE. METER) WRITE (IPRIN,1030) SIGWH(IS)
      IF (PUNITS(1) .EQ. METER) WRITE (IPRIN,1040) SIGWH(IS)
1030  FORMAT (/,42X,37HSEA STATE:  SIGNIFICANT WAVE HEIGHT =
    2 ,F6.2,7H FEET  )
1040  FORMAT (/,42X,37HSEA STATE:  SIGNIFICANT WAVE HEIGHT =
    2 ,F6.2,7H METERS)
      WRITE (IPRIN,1050) TMODAL(LT)
1050  FORMAT (54X,19HMODAL WAVE PERIOD =,F4.0,8H SECONDS)
      IF (NSVRSP .EQ. 5) GO TO 60
      NP = NSVRSP - 5
      NP = NP / 2
      WRITE (IPRIN.1025)
1025  FORMAT (//,54X,16HPOINT LOCATIONS:)
      DO 50 IP=1,NP
      WRITE (IPRIN,1026) IP,(PTNAME(I,IP),I=1,8),XPTLOC(IP),
    2 YPTLOC(IP),ZPTLOC(IP)
1026  FORMAT (22X,1HP.I1,3H-  ,8A4,2X,5HXFP =,F7 2,2X,5HYCL =,F7.2,2X,
    2 5HZBL =,F7.2)
 50   CONTINUE
 60   CONTINUE
      WRITE (IPRIN,1055) (STATNM(I),I=1,3)
1055  FORMAT (/,40X,3A4,39H VALUE / ENCOUNTERED MODAL PERIOD (TOE))
      WRITE (IPRIN,1060) ((RSPNME(I,IR),I=1,2),IR=1,NSVRSP)
1060  FORMAT (/,48X,32HMAXIMUM RESPONSES AND CONDITIONS,/,1X,
    2 130(1H-),//,14H RESPONSE     ,13(4X,A4,A1))
      DO 310 IR=1,NSVRSP
      IV = RV(IR)
      IH = RH(IR)
      TEMV(IR) = VK(IV)
      TEMH(IR) = HDNG(IH)
      TEMR(IR) = RSV(IR,IR)
      IF (IR .GT. 5) TEMR(IR) = TEMR(IR) * 100
      TEMT(IR) = TOE(IR,IR)
      IF (TEMT(IR) .GE. 99) TEMT(IR) = 99
 310  CONTINUE
      WRITE (IPRIN,1070) (TEMR(IR),TEMT(IR),IR=1,NSVRSP)
1070  FORMAT (/,14H (MAX.RSV)/TOE,13(1X,F5.2,1H/,I2))
      WRITE (IPRIN,1080) (TEMV(IR),IR=1,NSVRSP)
1080  FORMAT (17H AT SPEED (KNOTS),F6.1,12F9.1)
      WRITE (IPRIN,1090) (TEMH(IR),IR=1,NSVRSP)
1090  FORMAT (17H AT HEADING (DEG),F6.0,12F9.0)
      WRITE (IPRIN,1100) ((RSPNME(I,JR),I=1,2),JR=1,NSVRSP)
1100  FORMAT (//,54X,20HASSOCIATED RESPONSES,/,1X,130(1H-),//,
    2 15H MAX.   SPEED /,/,15H RESPN. HEADING,3X,A4,A1,12(4X,A4,A1))
      WRITE (IPRIN,1110)
1110  FORMAT (1X)
      DO 330 IR=1,NSVRSP
      IV = RV(IR)
      IH = RH(IR)
      MV = VK(IV) + .001
      MH = HDNG(IH) + .001
      IF (IR.EQ.6 .OR. IR.EQ.8 .OR. IR.EQ.10 .OR. IR.EQ.12)
    2 WRITE (IPRIN,1110)
      DO 320 JR=1,NSVRSP
      TEMR(JR) = RSV(JR,IR)
      IF (JR .GT. 5) TEMR(JR) = TEMR(JR) * 100
      TEMT(JR) = TOE(JR,IR)
      IF (TEMT(JR) .GE. 99) TEMT(JR) = 99
 320  CONTINUE
```

```
      WRITE (IPRIN,1120) (RSPNME(I,IR),I=1,2),MV,MH,(TEMR(JR),TEMT(JR),
    2 JR=1,NSVRSP)
 1120 FORMAT (1X,A4,A1,2X,I2,1H/,I3,13(F6.2,1H/,I2))
  330 CONTINUE
      WRITE (IPRIN,1130)
 1130 FORMAT (//,2X,42HNOTES: 1) RESPONSES ARE IN PHYSICAL UNITS:,/,
    2 22X,50HHEAVE AND SWAY ARE IN WAVE HEIGHT UNITS;   PITCH, ,
    2 29HROLL, AND YAW ARE IN DEGREES;,/,22X,23HAND THE POINT VERTICAL,
    2 53HAND LATERAL ACCELERATIONS ARE IN UNITS OF  G-S * 100.)
      WRITE (IPRIN,1140)
 1140 FORMAT (9X,51H2) POINT LOCATIONS:    XFP IS IN STATION NUMBERS;
    2 ,37HYCL AND ZCL ARE IN WAVE HEIGHT UNITS.)
      WRITE (IPRIN,1150)
 1150 FORMAT(9X,52H3) HEADING CONVENTION: 0 DEG=HEAD, 90 DEG=STBD BEAM,
    2 ,24H 180 DEG=FOLLOWING SEAS.)
  400 CONTINUE
  500 CONTINUE

      CLOSE (UNIT=SEVFIL)

      RETURN
      END

C DECK SKFRSP
      FUNCTION SKFRSP (WE,LPP,V,SFD)

      REAL LPP

      SKFRSP = SFD*(1. + 4.1*V/(WE*LPP))

      RETURN
      END

C DECK SKLIFT
      SUBROUTINE SKLIFT

      COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
    2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
    2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
    2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
    2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
    2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2)
    2,SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
    2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
    2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
    2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
    2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
    1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(17,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
    2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
    1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
    2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
    2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
    2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(5,10),LPP,BEAM,DBLWL,TLCB,
    2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
    2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
    4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
    5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
    2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
        1 RHOF,GNUS,GNUF,FTMETR

          COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
        2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
        2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
        2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
        2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
        2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
        2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
        2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
        2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
        2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
        2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
        2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
        2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
        2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
        2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
          REAL RDBLK(2692)
          EQUIVALENCE (PSUR(1),RDBLK(1))

          REAL LCS,MCHORD

          IF (NSKSET .EQ. 0) GO TO 20
          EN = 0
          STASPC = LPP/20
          DO 10 K=1,NSKSET
          XSKF - LCB - SKFLS(K)*STASPC
          XSKAU = LCB - SKAUS(K)*STASPC
          XSKAL = LCB - SKALS(K)*STASPC
          YSKG = SKHB(K)
          ZSKF = SKFLWL(K) - (DBLWL+VCG)
          ZSKAU = SKAUWL(K) - (DBLWL+VCG)
          ZSKAL = SKALWL(K) - (DBLWL+VCG)
          Q = SKTMAG(K)
          GAMMA = - 90
          SPAN = ZSKAU - ZSKAL
          MCHORD = (XSKF - XSKAL)/2

*       area

          AREA = SPAN*MCHORD

*       center of pressure

          XCP = XSKAL + (XSKF - XSKAL)/3
          YCP = YSKG
          ZCP = ZSKF + (ZSKAU - ZSKF)/6

*       moment arm

          GAM = GAMMA*DEGRAD
          YHAT = YCP*COS(GAM) + ZCP*SIN(GAM)

*       effective aspect ratio

          EAR = 2*SPAN/MCHORD

*       lift curve slope

          LCS = (PI/2)*EAR
          SQ(K) = Q
          SSPAN(K) = SPAN
          SMNCHD(K) = MCHORD
          SAREA(K) = AREA
          SXCP(K) = XCP
          SYCP(K) = YCP
          SZCP(K) = ZCP
          SGAMMA(K) = GAMMA
          SYHAT(K) = YHAT
          SEAR(K) = EAR
          SLCS(K) = LCS
          EN = EN + Q*(RHO/2)*AREA*LCS*YHAT*YHAT*WPHI*ENCON
```

```
   10 CONTINUE
   20 CONTINUE
      DO 30 IV=1,NVK
      ENSL(IV) = 0
      IF (NSKSET .GT. 0) ENSL(IV) = EN*VFS(IV)
   30 CONTINUE

      RETURN
      END

C DECK SKNFRC
      SUBROUTINE SKNFRC

      COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
     2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(6)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCE,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
     2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
     2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
     2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
     2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
     2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
     2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
     2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
     2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
     2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
     2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
     2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
     2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
     2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
     2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
      REAL RDBLK(2692)
      EQUIVALENCE (PSUR(1),RDBLK(1))

      DATA RNT/3.E5/

      DO 10 IA=1,NRANG
      DO 10 IS=1,NSIGMA
      SHPDMP(IS,IA) = 0
   10 CONTINUE
      DO 40 K=1,NSTATN
      IF (NOFSET(K) .LT. 2) GO TO 40
      RS = 1./PI*((0.887+0.145*CAK(K))*(1.7*ABS(DK(K))+CAK(K)*2*BMK(K))
     2 + 2.*VCG)
```

```
      CON = 4./(3.*PI)*RHO*PSUR(K)*RS**3
      DO 30 IA=1,NRANG
      DO 20 IS=1,NSIGMA
      PERE = TPI/SIGMA(IS)
      RN = (3.22*(RS*RANG(IA))**2 / (PERE*GNU)) * REYSCL

*    laminar flow

      CF = 1.328/SQRT(RN)

*    turbulent flow

      IF (RN .GE. RNT) CF = CF + 0.014*RN**(-0.114)
      STADMP(IS) = CON*SIGMA(IS)*RANG(IA)*CF
      STADMP(IS) = SIGMA(IS)*STADMP(IS)
      SHPDMP(IS.IA) = SHPDMP(IS,IA) + STADMP(IS)
   20 CONTINUE
   30 CONTINUE
   40 CONTINUE
      DO 50 IA=1,NRANG
      CALL SPFIT (SIGMA,SHPDMP(1,IA),SFELM(1,1,IA),NSIGMA)
      ENSFO = ENCON*REVAL (SFELM(1,ISIGMA,IA),WTSI)
      DO 45 IV=1,NVK
      ENSF(IV,IA) = SKFRSP (WPHI,LPP,VFS(IV),ENSFO)
   45 CONTINUE
   50 CONTINUE

      RETURN
      END

C DECK SLENTH
      SUBROUTINE SLENTH (AS,K)

      CHARACTER*(*) AS
      L=LEN(AS)
      K=L+1
      DO 10 M=1,L
      K=K-1
      IF (AS(K:K).NE.CHAR(32)) GO TO 20    ! Test for trailing blanks
   10 CONTINUE
   20 CONTINUE

      RETURN
      END

C DECK SMP93   -   Standard Ship Motion Program (SMP93)

      PROGRAM SMP93


*                Standard Ship Motion Program (SMP93)
*                      for Personal Computers


*                Operating system MS-DOS Version 4.01
*                  FORTRAN 77  using Lahey Fortran
*                   Overlay linking using PLINK86


*               Hull plot and Speed Polar/Density plots
*                      done in separate programs
*                    using HALO graphics language


       COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
     2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
     2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
     2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
     2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
     2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2)
     2,SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

  COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
  REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
  INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

  COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
  CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
  INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
  REAL KG

  COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
  INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
  REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

  COMMON /FINCON/ IACTFN,IFCLCS,FGAIN(8),FK(3),FA(3),FB(3),
2 FCLCS(8,2)

  COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
  INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
  CHARACTER*4 TITLE(20)
  REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

  COMMON /HULL/ A26

  COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
  INTEGER LPFIDX,LRMIDX,LSVIDX
  REAL PFIDX(235),RMIDX(183),SVIDX(3)

  COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
  INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

  COMMON /LOADS/ NLOADS,SWGHT(25),SMASS(25),XLDSTN(10),XLDXPT(25),
2 LSTATN(25)

  COMMON /PELEM/ PELEM
  COMPLEX PELEM(4,1000)

  COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
  COMPLEX II
  CHARACTER*4 PUNITS(2)
  REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

  COMMON /RDGEO/ BKLEN,WBKMAX,DLBKEL(25),SRBS(25),PHIS(25),CPS(25),
2 BKT(25),RKS(25),SSTR(25)
```

```
      COMMON /RESPN/ NRESP,IPOINT(182),IMOTN(182),ITYPE(182),
     2 ILIN(182),ISYM(182)
      LOGICAL ILIN,ISYM

      COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
     2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
     2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
     2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
     2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
     2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
     2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
     2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
     2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
     2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
     2 STADMP(10) SPEDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
     2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
     2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
     2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
     2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
      REAL RDBLK(2692)
      EQUIVALENCE (PSUR(1),RDBLK(1))

      COMMON /SEVERE/ NRSIND,RSINDX,NSWIND,SWINDX,RSVTOE,RV,RH
      REAL RSINDX(14),SWINDX(5),RSVTOE(402)
      INTEGER RV(13),RH(13)

      COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
     2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
     2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
     2 LSHIPS,LTITLES
      CHARACTER*160 AS
      CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
      CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
      CHARACTER SHIPS*6,VARS*2,CYCLS*2
      INTEGER*2 OPTION

      COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
      LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

      COMMON /STELEM/ STELEM
      COMPLEX STELEM(4,9,250)

      COMMON /TELEM/ TELEM
      COMPLEX TELEM(4,9,10)

      COMMON /TWOD/ YY, ZZ, ENN, ISTA
      INTEGER ISTA
      REAL YY(10,25),ZZ(10,25),ENN(4,10,25)

      COMMON /WGHTS/ WTDL,NORM
      REAL WTDL(10,25),NORM(4,10,25)

      CHARACTER*20 DS,TS,ES,T1S,T2S

* START


*  set underflow to zero
*     CALL UNDERO (.TRUE.)
      CALL UNDFL (.TRUE.)

      AS='CLS'
      CALL SYSTEM (AS)

      CALL PRELIM

      CALL RDSMPSYS

      FIS = SOS(1:LSOS)//'.TEX'
      OPEN (TEXFIL,FILE=FIS,FORM='FORMATTED',STATUS='UNKNOWN')
      AS = '(/19X,"STANDARD SHIP MOTION PROGRAM, SMP93"/25X,'//
    2 '"FOR PERSONAL COMPUTERS")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

      AS = '(/28X,"DTRC   CODE 1561")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

      CALL DATE (DS)
      AS = '(//28X,"DATE = ",A20)'
      WRITE (*,AS) DS
      WRITE (TEXFIL,AS) DS

      CALL TIME (TS)
      T1S=TS
      AS = '(/28X,"TIME = ",A8)'
      WRITE (*,AS) TS
      WRITE (TEXFIL,AS) TS

      AS = '(//2X,"Running - ")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

      AS = '(//"  CALL INPUT")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

      CALL INPUT

      CALL TIME (T2S)
      CALL ELTIME (T1S,T2S)
      T1S=T2S

      IF (OPTN .EQ. 1) GO TO 10

      AS = '(//"  CALL REGWAV")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

      CALL REGWAV

      CALL TIME (T2S)
      CALL ELTIME (T1S,T2S)
      T1S=T2S

      AS = '(//"  CALL IRGSEA")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

      CALL IRGSEA

      CALL TIME (T2S)
      CALL ELTIME (T1S,T2S)
      T1S=T2S

      AS = '(//"  CALL OUTPUT")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

      CALL OUTPUT

      CALL TIME (T2S)
      CALL ELTIME (T1S,T2S)

* QUIT

   10 CONTINUE
```

```
      AS = '(//2X,"Finished ! ")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

      CALL TIME (ES)
      CALL ELTIME (TS,ES)

      CLOSE (UNIT=TEXFIL)
      CLOSE (UNIT=IPRIN)

      STOP
      END

C DECK SOLVE
      SUBROUTINE SOLVE (N,COFF,EXC,MOTN,UL,IP,IPRIN)

*    This routine obtains a solution of the lateral or vertical
*    equations of motion.
*             W.G.MEYERS, DTNSRDC, 072977

      COMPLEX COFF,EXC,MOTN,UL
      INTEGER N,IP
      DIMENSION COFF(N,N),EXC(N),MOTN(N),UL(N,N)
      DIMENSION IP(N)

      CALL CDCOMP(N,N,COFF,UL,IP)
      IF (IP(N) .EQ. 0) WRITE (IPRIN,1000)
 1000 FORMAT (42H  SOLVE -- PROGRAM STOP.  MATRIX SINGULAR.)
      IF (IP(N) .EQ. 0) STOP
      CALL CSOLVE(N,N,UL,EXC,MOTN,IP)

      RETURN
      END

C DECK SPFIT
      SUBROUTINE SPFIT (X, Y, ELEMS, NPTS)

*        SPFIT created from SPLINE      E N HUBBLE      JUNE 19
*     fits cubic non-parametric spline segments
*              to set of real data points

*     INPUTS
*        X      =  array of real independent variables
*        Y      =  array of real dependent variables
*        NPTS   =  number of  (X,Y)  data points

*     RETURN
*        ELEMS  =  array of (NPTS-1) segments in following form
*                  ( Y(I), D(I), Y(I+1), D(I+1) )      ,  where
*           D   =  array of second derivatives at data points

*     arrays A,B,C are mainly sub diag., diagonal, and super diag.
*     D array is the right hand side of matrix equation
*     second derivatives at nodes are placed in D array after solution
*     solution technique is gaussian elimination
*     boundary conditions set by extrapolation of second derivatives

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

      DIMENSION X(NPTS),Y(NPTS),ELEMS(4,NPTS)
      DIMENSION A(100), B(100), C(100), D(100)

      N = NPTS
      NL1 = N - 1
      NL2 = N - 2
      DO 50 I=2,N
      IF (X(I) .GT. X(I-1))  GO TO 50
      WRITE (IPRIN,888) X(I-1),X(I)
      GO TO 88888
50    CONTINUE
      IF (N .LE. 100)  GO TO 100
      WRITE (IPRIN,999)
      N = 100
100   CONTINUE
      IF (N .GT. 2)  GO TO 125
      D(1) = 0.0
      D(2) = 0.0
      GO TO 375
125   CONTINUE
      IF (N .GT. 3)  GO TO 150
      YDD = 2.*((X(3)-X(2))*Y(1)+(X(2)-X(1))*Y(3)-(X(3)-X(1))*Y(2))
     2 /((X(3)-X(2))*(X(2)-X(1))*(X(3)-X(1)))
      D(1) = YDD
      D(2) = YDD
      D(3) = YDD
      GO TO 375
150   CONTINUE
      DO 200 I=1,N
      A(I) = 0.0
      B(I) = 0.0
      C(I) = 0.0
      D(I) = 0.0
200   CONTINUE

*       set up matrices(a tridiagonal structure)

      A(1) = (X(3)-X(2))/(X(3)-X(1))
      C(1) = 2.0
      B(1) = 1.0 - A(1)
      D(1) = 6.0*((Y(3)-Y(2))/(X(3)-X(2))-(Y(2)-Y(1))/
     1 (X(2)-X(1)))/(X(3)-X(1))
      H = X(3) - X(2)
      DO 250 I=3,NL1
      HP = X(I+1) - X(I)
      C(I) = HP / (H+HP)
      B(I) = 2.0
      A(I) = 1.0 - C(I)
      D(I) = 6.0*((Y(I+1)-Y(I))/HP-(Y(I)-Y(I-1))/H)/(HP+H)
      H = HP
250   CONTINUE

*       set boundary conditions

      C(2) = (X(2)-X(1))/(X(3)-X(2))
      A(2) = 1.0
      B(2) = -1.0-C(2)
      D(2) = 0.0
      C(2) = -A(2)*A(1)/B(1) + C(2)
      C(N) = (X(N)-X(N-1))/(X(N-1)-X(N-2))
      A(N) = -1.0 - C(N)
      B(N) = 1.0
      D(N) = 0.0

*       solve equations

      II = 1
      DO 300 I=1,NL2
      I1 = I + 1
      I2 = I + 2
      AUGH = ABS (B(I))
      IF (AUGH .LT. 1.0E-06)  GO TO 275
      CONST = A(I1) / B(I)
      B(I1) = B(I1) - CONST*C(I)
      D(I1) = D(I1) - CONST*D(I)
      IF (I .NE. NL2)  GO TO 300
      A(N) = A(N) - C(N)*C(I) / B(I)
      D(N) = D(N) - C(N)*D(I) / B(I)
      GO TO 300
275   CONTINUE
```

```
      II = I + 1
      D(I) = D(I) / C(I)
      D(I1) = D(I1) - B(I1)*D(I)
      B(I1) = A(I1)
      A(I1) = 0.0
      D(I2) = D(I2) - A(I2)*D(I)
      A(I2) = 0.0
      IF (I .NE. NL2)  GO TO 300
      A(N) = C(N)
300   CONTINUE
      DET = B(NL1)*B(N) - C(NL1)*A(N)
      STORE = D(N)
      D(N) = (B(NL1)*D(N) - D(NL1)*A(N)) / DET
      D(NL1) = (D(NL1)*B(N) - C(NL1)*STORE) / DET
      IP = 0
      DO 350 I=2,NL2
      JI = N - I
      IF (JI .EQ. IP)  GO TO 350
      IF (JI .EQ. II)  GO TO 325
      D(JI) = (D(JI)-C(JI)*D(JI+1))/B(JI)
      GO TO 350
325   CONTINUE
      IP = JI-1
      STORE = D(JI)
      D(JI) = D(IP)
      D(IP) = (STORE - C(IP)*D(JI+1))/B(IP)
350   CONTINUE
      D(1) = (D(1) - A(1)*D(3) - C(1)*D(2)) / B(1)

*       set up spline segments

375   CONTINUE
      DO 400 I=1,NL1
      I1 = I + 1
      ELEMS(1,I) = Y(I)
      ELEMS(2,I) = D(I)
      ELEMS(3,I) = Y(I1)
      ELEMS(4,I) = D(I1)
400   CONTINUE
99999 CONTINUE

      RETURN
88888 CONTINUE

      STOP
888   FORMAT ('0 SPFIT -- X VALUES NOT ASCENDING', 2E16.8)
999   FORMAT ('0 SPFIT -- NPTS EXCEEDS 100. ONLY 99 SEGMENTS RETURNED')

      END

C DECK SPINT2
      SUBROUTINE SPINT2 (SEGS, NSEGS, AREA, NS, TS, NE, TE, IWAY)

*     evaluates the integral of a function given as a parametric spline

*     INPUTS
*        SEGS  =  spline segements generated by SPLNT2
*        NSEGS =  number of spline segments
*        NS    =  index of segment  for start of integration
*        TS    =  t parameter  for start of integration
*        NE    =  index of segment for end of integration
*        TE    =  t parameter  for end of integration
*        IWAY  =  -1 ,  if integral of y dx is to be evaluated
*        IWAY  =   0 ,  if integral of x dy is to be evaluated

*     RETURN
*        AREA  =  INTEGRAL (AREA UNDER CURVE)  FROM (NS+TS) TO (NE+TE)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

       DIMENSION SEGS(8,NSEGS),CC(14),T(2),A(2)

       AREA = 0.0
       IF (NS.GE.1 .AND. NS.LE.NSEGS)  GO TO 100
       WRITE (IPRIN,991) NS
       GO TO 99999
100    CONTINUE
       IF (NE.GT.NS .AND. NE.LE.NSEGS)  GO TO 150
       WRITE (IPRIN,992) NE
       GO TO 99999
150    CONTINUE
       IF (TS.GE.0.0 .AND. TS.LE.1.0)  GO TO 200
       WRITE (IPRIN,993) TS
       GO TO 99999
200    CONTINUE
       IF (TE.GE.0.0 .AND. TE.LE.1.0)  GO TO 250
       WRITE (IPRIN,994) TE
       GO TO 99999
250    CONTINUE
       IF (IWAY .EQ. 0) GO TO 350
       K = 1
       J = 2
       GO TO 400
350    CONTINUE
       K = 2
       J = 1
400    CONTINUE
       J2 = J + 2
       J4 = J + 4
       J6 = J + 6
       K4 = K + 4
       K8 = K + 8
       K10 = K + 10
       DO 600 I=NS,NE
       T(1) = 0.0
       T(2) = 1.0
       IF (I .EQ. NS)  T(1) = TS
       IF (I .EQ. NE)  T(2) = TE
       CALL CUBCO2 (SEGS(1,I), CC)
       DD1 = (CC(J)*CC(K8)) / 6.0
       DD2 = (CC(J)*CC(K10) + CC(J2)*CC(K8)) / 5.0
       DD3 = (CC(J )*CC(K4) + CC(J2)*CC(K10) + CC(J4)*CC(K8)) / 4.0
       DD4 = (CC(J2)*CC(K4) + CC(J4)*CC(K10) + CC(J6)*CC(K8)) / 3.0
       DD5 = (CC(J4)*CC(K4) + CC(J6)*CC(K10)) / 2.0
       DD6 = CC(J6)*CC(K4)
       DO 550 L=1,2
       IF (T(L) .GT. 0.0)  GO TO 450
       A(L) = 0.0
       GO TO 550
450    CONTINUE
       IF (T(L) .LT. 1.0)  GO TO 500
       A(L) = DD1 + DD2 + DD3 + DD4 + DD5 + DD6
       GO TO 550
500    CONTINUE
       A(L) = ((((( DD1 * T(L) + DD2) * T(L) + DD3) * T(L) + DD4)
     2 * T(L) + DD5) * T(L) + DD6) * T(L)
550    CONTINUE
       AREA = AREA + A(2) - A(1)
600    CONTINUE
99999  CONTINUE

       RETURN

991    FORMAT ('0 SPINT2 -- NS =', I5, '    OUT OF RANGE' )
992    FORMAT ('0 SPINT2 -- NE =', I5, '    OUT OF RANGE' )
993    FORMAT ('0 SPINT2 -- TS =', E12.5,  '    OUT OF RANGE' )
994    FORMAT ('0 SPINT2 -- TE =', E12.5,  '    OUT OF RANCE' )

       END
```

```
C DECK SPINTG
      SUBROUTINE SPINTG (XA, XB, X, NPTS, ELEMS, A, CINTG, SINTG)
*         SPINTG created from SUMSPL and SPLFIT
*      evaluates the integral of a real function defined by
*                non-parametric spline segments
*      INPUTS
*         XA      =  lower limit of integration
*         XB      =  upper limit of integration
*         X       =  array of independent variables
*         NPTS    =  number of values in x-array
*         ELEMS   =  non-parametric spline segments generated by SPFIT
*         A       =  constant for specific integral to be evaluated
*      RETURNS
*         CINTG  =  INTEGRAL OF  F(X) * COS(A*X)
*         SINTG  =  INTEGRAL OF  F(X) * SIN(A*X)
*         IF  A = 0.0 ,  THEN  CINTG = INTEGRAL OF F(X),  AND  SINTG = 0.

       DIMENSION X(NPTS),ELEMS(4,NPTS)

       CINTG = 0.0
       SINTG = 0.0
       CALL SPLVAL (X, NPTS, ELEMS, XA, YA, SA, IA)
       CALL SPLVAL (X, NPTS, ELEMS, XB, YB, SB, IB)
       A2 = A * A
       A3 = A * A2
       A4 = A * A3
       DO 500 I=IA,IB
       IF (I .GT. IA) GO TO 100
       X1 = XA
       X2 = X(I+1)
       Y1 = YA
       Y2 = ELEMS(3,I)
       S1 = SA
       S2 = ELEMS(4,I)
       GO TO 300
100    CONTINUE
       IF (I .LT. IB) GO TO 200
       X1 = X(I)
       X2 = XB
       Y1 = ELEMS(1,I)
       Y2 = YB
       S1 = ELEMS(2,I)
       S2 = SB
       GO TO 300
200    CONTINUE
       X1 = X(I)
       X2 = X(I+1)
       Y1 = ELEMS(1,I)
       Y2 = ELEMS(3,I)
       S1 = ELEMS(2,I)
       S2 = ELEMS(4,I)
300    CONTINUE
       XX = X2 - X1
       IF (A .NE. 0.0) GO TO 400
       SEGINT = (Y2+Y1) * XX / 2.  -  (S2+S1) * XX**3 / 24.
       CINTG = CINTG + SEGINT
       GO TO 500
400    CONTINUE
       ZAA = (S2-S1) / (XX * 6.)
       ZBB = S1 / 2.
       ZCC = (Y2-Y1) / XX  -  (S2 + 2.*S1) * XX / 6.
       AXX = A * XX
       E = SIN (AXX)
       F = COS (AXX)
       XX2 = XX * XX
       XX3 = XX * XX2
       P     = (3.*A2*XX2 - 6.) / A4
       Q     = (A2*XX3 - 6.*XX) / A3
       AA1 = F*P + E*Q + 6./A4
       AA2 = E*P - F*Q
       PP  = (2.*XX) / A2
       QQ  = (A2*XX2 - 2.) / A3
       BB1 = F*PP + E*QQ
       BB2 = E*PP - F*QQ - 2./A3
       XXA = XX / A
       CC1 = (F-1.)/A2 + E*XXA
       CC2 =       E/A2 - F*XXA
       DD1 =       E/A
       DD2 = (1.-F)/A
       AX1 = A * X1
       VV = COS (AX1)
       UU = SIN (AX1)
       PPP = (AA1*ZAA + BB1*ZBB + CC1*ZCC + DD1*Y1)
       QQQ = (AA2*ZAA + BB2*ZBB + CC2*ZCC + DD2*Y1)
       SISEG = UU*PPP + VV*QQQ
       CISEG = VV*PPP - UU*QQQ
       CINTG = CINTG + CISEG
       SINTG = SINTG + SISEG
500    CONTINUE

       RETURN
       END

C DECK SPLNAR
       SUBROUTINE SPLNAR (P,NPTS,SPAREA,PSEGS,NS)

       COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
       INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

       DIMENSION P(2,10),PSEGS(8,9)
       DIMENSION NDI(2),ENDI(2,2)

       DATA ZERO,ONE /0.0,1.0/
       DATA NDI,ENDI /2*1,4*0.0/

       CALL SPLNT2 (PSEGS,P,NPTS,NDI,ENDI)
       CALL SPINT2 (PSEGS,NS,SPAREA,1,ZERO,NS,ONE,0)

       RETURN
       END

C DECK SPLNFT
       SUBROUTINE SPLNFT

*      routine used to write offsets to HPLFIL for graphics

       COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
     2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
     2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
     2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
     2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
     2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
     2 STATNM,STATIS
       CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
       INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
     2 FBNUMB,PTNUMB,ORGOPTN
       REAL KG

       COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
       INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
       CHARACTER*4 TITLE(20)
       REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

       COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
       INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

       COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
       COMPLEX II
       CHARACTER*4 PUNITS(2)
       REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

       COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
     2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
     2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
     2 LSHIPS,LTITLES
       CHARACTER*160 AS
       CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
       CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
       CHARACTER SHIPS*6,VARS*2,CYCLS*2
       INTEGER*2 OPTION

       DIMENSION P(2,10),PSEGS(8,9),CC(14),AY(900),AZ(900),
     2 HFB(10,25),WTR(10,25),NDI(2),ENDI(2,2)
       CHARACTER*6 SNAME(6)
       CHARACTER*80 ATITL
       CHARACTER STSP*30

       DATA SNAME /'YFWD','ZFWD','YAFT','ZAFT','HLFBTH','WTRLNE'/
       DATA NDI,ENDI /2*1,4*0.0/

       DO 30 K=1,NSTATN
       NPTS = NSOFST(K)
       DO 10 I=1,NPTS
       HFB(I,K) = HLFBTH(I,K)
       WTR(I,K) = WTRLNE(I,K)
  10   CONTINUE
       IF (NPTS.EQ. 1 .AND. STATN(K).GT.10.0) HFB(1,K) = - HFB(1,K)
       NPT = 10 - NPTS
       IF (NPT .EQ. 0) GO TO 30
       DO 20 I=1,NPT
       IPT = I + NPTS
       HFB(IPT,K) = HFB(NPTS,K)
       WTR(IPT,K) = WTR(NPTS,K)
  20   CONTINUE
  30   CONTINUE
       DX = LPP/20
       WRITE(STSP,1000) DX,PUNITS(1),PUNITS(2)
 1000  FORMAT ('STATION SPACING =',F6.2,1X,A4,A2)
       WRITE(ATITL,1010) TITLE
 1010  FORMAT (20A4)

*   open file for hull offset plotting

       FIS = SDS(1:LSDS)//'.HPL'
       OPEN (UNIT=HPLFIL,FILE=FIS,STATUS='UNKNOWN')

       WRITE (HPLFIL,1020) ATITL
 1020  FORMAT (A80)
       WRITE (HPLFIL,1030) STSP
 1030  FORMAT (A30)

       NOS = 10
       L = 0
```

```
      KOUNT = 0
   40 IK = KOUNT + 1
      DO 100 K=IK,NSTATN
      KOUNT = KOUNT + 1
      NPTS = NSOFST(K)
      IF (NPTS .EQ. 1) GO TO 100
      L = L + 1
      AY(L) = 0.
      AZ(L) = WTR(NOS,K) - DRAFT
      DO 50 J=1,NOS
      IF (STATN(K) .GT. 10.0) HFB(J,K) = - HFB(J,K)
      WTR(J,K) = WTR(J,K) - DRAFT
      P(1,J) = HFB(J,K)
      P(2,J) = WTR(J,K)
   50 CONTINUE
      NS = NOS - 1
      CALL SPLNT2 (PSEGS,P,NOS,NDI,ENDI)
      DO 70 J=1,NS
      CALL CUBCO2 (PSEGS(1,J),CC)
      NT = 7
      DT = 1./(NT-1)
      DO 60 I=1,NT
      L = L + 1
      T = (I-1)*DT
      T2 = T*T
      T3 = T*T2
      AY(L) = CC(1)*T3 + CC(3)*T2 + CC(5)*T + CC(7)
      AZ(L) = CC(2)*T3 + CC(4)*T2 + CC(6)*T + CC(8)
   60 CONTINUE
   70 CONTINUE
      IF (STATN(K) .EQ. 10.0) GO TO 110
  100 CONTINUE

      WRITE (HPLFIL,1040) SNAME(3),SNAME(4)
      WRITE (HPLFIL,1050) L
      DO 210 I=1,L
      WRITE (HPLFIL,1060) AY(I),AZ(I)
  210 CONTINUE
      GO TO 120

  110 WRITE (HPLFIL,1040) SNAME(1),SNAME(2)
 1040 FORMAT (A6,4X,A6)
      WRITE (HPLFIL,1050) L
 1050 FORMAT (2I5)
      DO 220 I=1,L
      WRITE (HPLFIL,1060) AY(I),AZ(I)
 1060 FORMAT (10F7.2)
  220 CONTINUE
      L = 0
      GO TO 130

  120 WRITE (HPLFIL,1040) SNAME(5),SNAME(6)
      WRITE (HPLFIL,1050) NOS,NSTATN
      DO 230 K=1,NSTATN
      WRITE (HPLFIL,1060) (HFB(I,K),I=1,NOS)
      WRITE (HPLFIL,1060) (WTR(I,K),I=1,NOS)
  230 CONTINUE
  130 IF (KOUNT .LT. NSTATN) GO TO 40

      CLOSE (UNIT=HPLFIL)

      RETURN
      END

C DECK SPLNT2
      SUBROUTINE SPLNT2 ( SEGS, P, NP, NDI, ENDI )

*         SPLNT2 created from SPLNT ( NAVSEC-N000 ) - A M REED JULY 1976
*       fits cubic parametric spline segments through set of data points

*       INPUTS
*          P          =  array of (X,Y) points
*        NP        =  number of points
*        NDI(1)    =  1,  if initial slope not specified at first point
*        NDI(1)    =  2,  if initial slope is  specified at first point
*        NDI(2)    =  1,  if initial slope not specified at final point
*        NDI(2)    =  2,  if initial slope is  specified at final point
*        ENDI(1,1) =  DX/DT  at first point  -- not required if NDI(1)=1
*        ENDI(2,1) =  DY/DT  at first point  -- not required if NDI(1)=1
*        ENDI(1,2) =  DX/DT  at final point  -- not required if NDI(2)=1
*        ENDI(2,2) =  DY/DT  at final point  -- not required if NDI(2)=1

*     RETURNS
*        SEGS  =  array of (NP-1) segments in endpoint/tangent form
*                 X(I),Y(I),DX(I),DY(I),X(I+1),Y(I+1),DX(I+1),DY(I+1)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

      DIMENSION SEGS(8,NP),P(2,NP),NDI(2),ENDI(2,2)
      DIMENSION DS(2,70),INDEX(70),R1(70),R2(70),R3(70),R4(70),
     2 CS(70),T10(2),T21(2)

      DATA T10 / 1.0, 0.0 /,
     1     T21 / 2.0, 1.0 /

*        initialize segs array.  determine deltas, chord lengths and
*           indices of non-zero length segments.

      M = 1
      N = NP
      N1 = N - 1
      IF (N1 .LE. 69) GO TO 1000
      N1 = 69
      WRITE (IPRIN,999)
1000  CP = 0.0
      DO 1120 J = 1, N1
      INDEX(J) = J
      C = 0.0
      DO 1100 I = 1, 2
      P1 = P(I,J)
      P2 = P(I,J+1)
      DELTA = P2 - P1
      C = C + DELTA*DELTA
      DS(I,M) = 3.0*DELTA
      SEGS(I,J) = P1
      SEGS(I+4,J) = P2
      SEGS(I+2,J) = DELTA
      SEGS(I+6,J) = DELTA
1100  CONTINUE
      IF (C .LE. 0.000001) GO TO 1110
      C = SQRT( C )
      CS(M) = C
      R1(M) = C
      R3(M) = CP
      INDEX(M) = J
      M = M + 1
      CP = C
1110  CONTINUE
1120  CONTINUE
      N = M
      M = N - 1

*        check for degenerate case (only 2 points)

      IF (N .GT. 2) GO TO 1300

*           degenerate case.  set single segment tangent vectors.
```

```
      J = INDEX(1)
      C = CS(1)
      DO 1240 I = 1, 2
      IF ( NDI(1) .GT. 1 ) SEGS(I+2,J) = ENDI(I,1)*C
      IF ( NDI(2) .GT. 1 ) SEGS(I+6,J) = ENDI(I,2)*C
1240  CONTINUE
      GO TO 99999
1300  CONTINUE

*          set end conditions of tri-diagonal matrix

      I = NDI(1)
      R2(1) = T21(I)
      R3(1) = T10(I)
      I = NDI(2)
      R1(N) = T10(I)
      T2 = T21(I)

*          solve matrix for tangent vectors

      DO 1340 I = 1, 2
      R4(1) = DS(I,1)/CS(1)
      IF ( NDI(1) .GT. 1 ) R4(1) = ENDI(I,1)
      DO 1315 J = 2, M
      R = CS(J-1)/CS(J)
      R2(J) = 2.0*(CS(J) + CS(J-1))
      R4(J) = DS(I,J)*R + DS(I,J-1)/R
1315  CONTINUE
      R2(N) = T2
      R4(N) = DS(I,M)/CS(M)
      IF (NDI(2) .GT. 1) R4(N) = ENDI(I,2)
      DO 1330 J = 1, M
      R = R1(J+1)/R2(J)
      R2(J+1) = R2(J+1) - R3(J)*R
      R4(J+1) = R4(J+1) - R4(J)*R
1330  CONTINUE
      DN = R4(N)/R2(N)
      DO 1335 L = 1, M
      J = N - L
      K = INDEX(J)
      DJ = (R4(J) - R3(J)*DN)/R2(J)
      SEGS(I+2,K) = DJ*CS(J)
      SEGS(I+6,K) = DN*CS(J)
      DN = DJ
1335  CONTINUE
1340  CONTINUE
99999 CONTINUE

      RETURN

  999 FORMAT('0 SPLNT2 -- NP EXCEEDS 70.  ONLY 69 SEGMENTS RETURNED.')

      END

C DECK SPLVAL
      SUBROUTINE SPLVAL (X, NPTS, ELEMS, X0, Y0, S0, IELM)

*        SPLVAL created from SPLFIT
*     evaluates a real non-parametric spline

*     INPUTS
*        X     =  array of independent variables
*        NPTS  =  number of values in x-array
*        ELEMS =  spline segments generated by SPFIT
*        X0    =  x-value at which spline is to be evaluated

*     RETURNS
*        Y0    =  F(X0)  =  y-value evaluated at x0
*        S0    =  second derivative evaluated at x0
*        IELM  =  index of spline segment containing x0
```

```
      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
    2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
    2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
    2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
    2 SPTFIL,LACFIL,LAEFIL

      DIMENSION X(NPTS),ELEMS(4,NPTS)

      N = NPTS
      IF (X0.GE.X(1) .AND. X0.LE.X(N))  GO TO 100
      WRITE (IPRIN,999) X0
      GO TO 99999
100   CONTINUE
      DO 200 I=2,N
      IF (X0 .GT. X(I)) GO TO 200
      GO TO 300
200   CONTINUE
300   CONTINUE
      I = I - 1
      XX = X(I+1) - X(I)
      X1 = X0 - X(I)
      X2 = X(I+1) - X0
      XX6 = XX * XX / 6.0
      Y1 = ELEMS(1,I)
      Y2 = ELEMS(3,I)
      S1 = ELEMS(2,I)
      S2 = ELEMS(4,I)
      Y0 =  (S1 * X2**3 + S2 * X1**3) / (6.0 * XX)  +
    2 ( (Y1 - S1*XX6) * X2 + (Y2 - S2*XX6) * X1 ) / XX
      S0 = (S1 * X2  +  S2 * X1) / XX
      IELM = I

      RETURN

99999 CONTINUE

      STOP

999   FORMAT ('0 SPLVAL -- EXTRAPOLATION NOT ALLOWED.  X0 =', E16.8)

      END

C DECK SPPLV2
    SUBROUTINE SPPLV2 (V, P, SEGS, NSEGS, PT, NINT, TINT, INT)

*       SPPLV2 created from LNPLI2 and LNPLI
*   finds intersection between a curve defined by a parametric spline
*         and a plane defined by a point and a direction vector

*   INPUTS
*       P(1)  =  X-COORDINATE OF POINT   USED TO DEFINE THE PLANE
*       P(2)  =  Y-COORDINATE OF POINT   USED TO DEFINE THE PLANE
*       V(1)  =  X-COMPONENT OF VECTOR PERPENDICULAR TO THE PLANE
*       V(2)  =  Y-COMPONENT OF VECTOR PERPENDICULAR TO THE PLANE
*       SEGS  =  SPLINE SEGMENTS IN ENDPOINT-TANGENT FORM, FROM SPLNT2
*       NSEGS =  NUMBER OF SPLINE SEGMENTS

*   RETURNS
*       PT(1) =  X-COORDINATE OF THE INTERSECTION
*       PT(2) =  Y-COORDINATE OF THE INTERSECTION
*       NINT  =  INDEX OF SEGMENT IN WHICH INTERSECTION LIES
*       TINT  =  VALUE OF T PARAMETER AT INTERSECTION
*       INT   =  1,  IF INTERSECTION FOUND AND WITHIN TOLERANCE
*       INT   =  2,  IF INTERSECTION NOT WITHIN TOLERANCE
*       INT   =  3,  IF NO INTERSECTION FOUND
*       INT   =  4,  IF SEGMENT LIES WITHIN THE PLANE

      DIMENSION V(2),P(2),SEGS(8,NSEGS),PT(2),CC(14),U(2)

      EQUIVALENCE (U1,U(1)), (U2,U(2)), (CC1,CC(1)), (CC2,CC(2)),
    1 (CC3,CC(3)), (CC4,CC(4)), (CC5,CC(5)), (CC6,CC(6)), (D,DPS)
```

```
       DATA  TOLER, IMAX  /  0.001,  10  /

       INT=1

*        unitize plane direction vector

       CALL VUNIT2 (U, S, V)

*        determine the segment number n which contains the intersection

       DO 140 N=1,NSEGS
       DPS=0.0
       DPE=0.0
       DO 1000 I = 1, 2
       DPS = DPS + (SEGS(I,N) - P(I))*U(I)
       DPE = DPE + (SEGS(I+4,N) - P(I))*U(I)
1000   CONTINUE

*        check if segment lies within plane.  if so, set int and return.

       IF ( ABS( DPS ) .GT. TOLER .OR.
     1 ABS( DPE ) .GT. TOLER ) GO TO 130
       INT=4
       GO TO 99999
130    CONTINUE

*        check if dot product changes sign within segment

       NSEG=N
       IF ( DPS*DPE .LT. 0.0 ) GO TO 200
       IF ( DPS*DPE .EQ. 0.0 ) GO TO 145
140    CONTINUE
       NSEG=NSEGS
       N=1
145    CONTINUE

*        check if intersection occurs at either end of line

       T=0.0
       DO 1170 J = 1, 5, 4
       DIST = 0.0
       DO 1150 I = 1, 2
       K = I + J - 1
       PT(I) = SEGS(K,N)
       DIST = DIST + (PT(I)-P(I)) * U(I)
1150   CONTINUE
       IF ( ABS(DIST) .LE. TOLER )  GO TO 1440
       N = NSEG
       T = 1.0
1170   CONTINUE

*        no intersection found.  set int and return.

       INT=3
       GO TO 99999
200    CONTINUE

*        fetch segment polynomial coefficients

       CALL CUBCO2 (SEGS(1,N), CC)

*        determine scalar polynomial coefficients

       A = CC1*U1 + CC2*U2
       B = CC3*U1 + CC4*U2
       C = CC5*U1 + CC6*U2
       A3=A*3.0
       B2=B*2.0

*        iterate for t at which the scalar polynomial becomes zero
```

```
      ITER=0
      T=DPS/(DPS-DPE)
300   CONTINUE
      FT=((A*T+B)*T+C)*T+D
      DT=FT/((A3*T+B2)*T+C)
      T=T-DT
      IF ( ABS( DT ) .LE. 0.0000001 ) GO TO 400
      ITER=ITER+1
      IF ( ITER .LE. IMAX ) GO TO 300
      IF ( ABS( FT ) .GT. TOLER ) INT = 2
400   CONTINUE

*       set intersection coordinates, n and t parameters

      DO 1420 I = 1, 2
      COORD = ((CC(I)*T + CC(I+2))*T + CC(I+4))*T + CC(I+6)
      IF ( ABS( COORD - P(I) ) .LE. TOLER ) CCORD = P(I)
      PT(I) = COORD
1420  CONTINUE
1440  CONTINUE
      NINT=N
      TINT=T
99999 CONTINUE

      RETURN
      END

C DECK T2DAMD
      SUBROUTINE T2DAMD (K,PHI2D,T2D,T3D)

*  calculates added mass and damping forces on a 2-d section given
*  the potentials

      COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
     2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(6),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      COMMON /WGHTS/ WTDL,NORM
      REAL WTDL(10,25),NORM(4,10,25)

      COMPLEX PHI2D(10,10,4),CTEMP,T2D(10,10),T3D(10,10)
      DIMENSION IDX(10),JDX(10)
      DIMENSION  T(25),ELEMS(4,25)

      DATA IDX/1,3,5,3,2,4,6,2,2,4/
      DATA JDX/1,3,5,5,2,4,6,4,6,6/

      NNODES=NOFSET(K)
      IF(NNODES.LE.0) RETURN
      DO 3 I=1,NSTATN
      T(I)=0.0
    3 CONTINUE
      T(K)=1.0
      CALL SPFIT ([illegible],ELEMS,NSTATN)
      CALL SPINTG ([illegible]),X(NSTATN),X,NSTATN,ELEMS,0.0,WTLI,DUM)
      DO 10 ISIGMA=1,NSIGMA
      DO 1  L=LMIN,LMAX
      CTEMP = (0.,0.)
      I=IDX(L)
      IN = I
      IF (I .EQ. 5) IN = 3
      IF (I .EQ. 6) IN = 2
      J=JDX(L)
      JP=J
      IF (J .EQ. 5)  JP=3
      IF (J .EQ. 6)  JP=2
      XFCTR=1.0
      IF (I .EQ. 5)  XFCTR=-XFCTR*X(K)
      IF (I .EQ. 6)  XFCTR= XFCTR*X(K)
      IF (J .EQ. 5)  XFCTR=-XFCTR*X(K)
      IF (J .EQ. 6)  XFCTR= XFCTR*X(K)
      DO 2 M=1,NNODES
      CTEMP = CTEMP + WTDL(M,K)*NORM(IN,M,K)*PHI2D(ISIGMA,M,JP)
    2 CONTINUE
      T2D(ISIGMA,L) = 2.0*II*RHO*SIGMA(ISIGMA)*XFCTR*CTEMP
      T3D(ISIGMA,L) = T3D(ISIGMA,L) + WTLI*T2D(ISIGMA,L)
    1 CONTINUE
   10 CONTINUE

      RETURN
      END

C DECK T3DAMD
      SUBROUTINE T3DAMD

      COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
     2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
     2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
     2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
     2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
     2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
     2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
     2 STATNM,STATIS
      CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
      INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
     2 FBNUMB,PTNUMB,ORGOPTN
      REAL KG

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

       COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
       INTEGER LPFIDX,LRMIDX,LSVIDX
       REAL PFIDX(235),RMIDX(183),SVIDX(3)

       COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
       INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

       COMMON /PELEM/ PELEM
       COMPLEX PELEM(4,1000)

       COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
       LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

       COMMON /STELEM/ STELEM
       COMPLEX STELEM(4,9,250)

       COMMON /TELEM/ TELEM
       COMPLEX TELEM(4,9,10)

       COMMON /WGHTS/ WTDL,NORM
       REAL WTDL(10,25),NORM(4,10,25)

       COMPLEX T3D(10,10),PHI2D(10,10,4)
       EQUIVALENCE (PELEM(1,1),T3D(1,1)),(PELEM(1,26),PHI2D(1,1,1))
       COMPLEX T2D(10,10)

       READ (SCRFIL) WTDL,NORM
       BACKSPACE SCRFIL
       IMMIN = 1
       IF (.NOT. VRT) IMMIN = 2
       IMMAX = 4
       IF (.NOT. LAT) IMMAX = 3
       IMDEL = 2
       IF (VRT .AND. LAT) IMDEL = 1
       LMIN = 1
       IF (.NOT. VRT) LMIN = 5
       LMAX = 10
       IF (.NOT. LAT) LMAX = 4
       DO 20 I=1,10
       DO 10 J=1,10
       T3D(I,J) = (0.0,0.0)
 10    CONTINUE
 20    CONTINUE
       DO 30 K=1,NSTATN
       NPT = NOFSET(K)
       IF (NPT .LT. 2) GO TO 30
       CALL RPHI2D (K,PHI2D)
       CALL T2DAMD (K,PHI2D,T2D,T3D)
       M = (K-1)*10
       DO 25 L=LMIN,LMAX
       M = M + 1
       CALL CPFIT (SIGMA,T2D(1,L),STELEM(1,1,M),NSIGMA)
 25    CONTINUE
 30    CONTINUE
       DO 40 L=LMIN,LMAX
       CALL CPFIT (SIGMA,T3D(1,L),TELEM(1,1,L),NSIGMA)
 40    CONTINUE
       REWIND COFFIL
       WRITE (COFFIL) TELEM
       REWIND COFFIL
       IF (RLDMPR .GT. 0) CALL AMDPRN (SIGMA,NSIGMA)

       RETURN
       END
```

```
C DECK TANAKA
      SUBROUTINE TANAKA

*  calculates coefficient C (=EDDY(K)) and RADIUS (=RGB(K))
*  for calculating eddy-making roll damping by the method of
*           TANAKA, J.ZOSEN KIOKAI, V. 109, 1961

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
     2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
     2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
     2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
     2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
     2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
     2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
     2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
     2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
     2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
     2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
     2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
     2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
     2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
     2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
      REAL RDBLK(2692)
      EQUIVALENCE (PSUR(1),RDBLK(1))

      DO 20 IA=1,NRANG
      DO 10 K=1,NSTATN
      EDDY(IA,K) = 0
      RGB(K) = 0.
      IF (NOFSET(K) .LT. 2) GO TO 10
      BLOCAL = BMK(K)
      TLOCAL = DK(K)
      ORG = TLOCAL - VCG
      IF (ITS(K) .EQ. 1) CALL SERD (K,RANG(IA),BLOCAL,TLOCAL,ORG,
     2 EDDY(IA,K),RGB(K))
      IF (ITS(K) .EQ. 2) CALL SERAB (K,RANG(IA),BLOCAL,TLOCAL,ORG,
     2 RD(K),EDDY(IA,K),RGB(K))
      IF (ITS(K) .NE. 3) GO TO 10

*    stations with skegs

      ORG = TLOCAL - VCG
      CALL SERE (BLOCAL,ORG,EDDY(IA,K),RGB(K))
   10 CONTINUE
   20 CONTINUE

      RETURN
      END

C DECK TEPEAK
      SUBROUTINE TEPEAK (NWEVN,WEVN,ERS,XTOE,TPI)
```

```
*    this routine obtains the period of max energy of an encounter
*    spectrum.
*          W.G.MEYERS, DTNSRDC, 072877

      DIMENSION WEVN(NWEVN),ERS(NWEVN)
      PEAK = 0.
      XTOE = TPI/WEVN(1)
      DO 10 I=1,NWEVN
      TE = TPI/WEVN(I)
      IF (ERS(I).GT.PEAK) XTOE = TE
      IF (ERS(I).GT.PEAK) PEAK = ERS(I)
   10 CONTINUE

      RETURN
      END

C DECK TFNFIT
      SUBROUTINE TFNFIT (RLANG,NRANG,RLANS,MOTL,JM,IW,CTFN)

      DIMENSION RLANG(8)
      COMPLEX MOTL(3,30,8),CANS(8),CELM(4,8),CTFN,CDUM

      IF (RLANS .GE. RLANG(1)) GO TO 10
      CTFN = MOTL(JM,IW,1)
      GO TO 40
   10 IF (RLANS .LE. RLANG(NRANG)) GO TO 20
      CTFN = MOTL(JM,IW,NRANG)
      GO TO 40
   20 DO 30 IA=1,NRANG
      CANS(IA) = MOTL(JM,IW,IA)
   30 CONTINUE
      CALL CPFIT (RLANG,CANS,CELM,NRANG)
      CALL CPLVAL (RLANG,NRANG,CELM,RLANS,CTFN,CDUM,IELM)
   40 CONTINUE

      RETURN
      END

C DECK TOE
      SUBROUTINE TOE (KREC,AOMGE,RAO1,RAO2,JA,IT,R,B2,NPREDH,NLCH,N1,
     2 N2,NBETA,DELBET,NWEVN,WEVN,IV,DATA)

      DIMENSION KREC(13),AOMGE(30,13),RAO1(30,8,13),RAO2(30,8,11),
     2 R(30),B2(35),WEVN(100),DATA(432),DUM1(30),DUM2(30),ARLC1(100),
     2 ARLC2(100),ARLC3(100),RLC(100,24)

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      INTEGER DELBET

      DO 50 IH=1,NMU
      HDNG = (IH-1)*DELBET
      I1 = N1 + IH
      I2 = N2 - IH
      IF (I2 .LE. 0) I2 = I2 + NBETA
      IF (KREC(IH) .GT. 0) GO TO 20
      DO 10 I=1,NWEVN
   10 RLC(I,I1) = 0.
      GO TO 50
   20 CALL PSPLC (NOMEGA,OMEGA,AOMGE(1,IH),VK(IV),HDNG,DEGRAD,GRAV,
     2 VKMETR,DUM1,DUM2,RAO1(1,JA,IH),S(1,IT),R,NWEVN,WEVN,ARLC1,ARLC2,
     2 ARLC3,RLC(1,I1))
       IF (KREC(IF) .EQ. 2) GO TO 40
       DO 30 I=1,NWEVN
   30  RLC(I,I2) = RLC(I,I1)
       GO TO 50
   40  KH = IH - 1
       CALL PSPLC (NOMEGA,OMEGA,AOMGE(1,IH),VK(IV),HDNG,DEGRAD,GRAV,
     2 VKMETR,DUM1,DUM2,RAO2(1,JA,KH),S(1,IT),R,NWEVN,WEVN,ARLC1,ARLC2,
     2 ARLC3,RLC(1,I2))
   50  CONTINUE

       L = 0
       DO 60 IPH=1,NPREDH
       CALL PSPSC (NWEVN,WEVN,RLC,NBETA,B2,NLCH,IPH,ARLC1,ARLC2,TOELC,
     2 TOESC,TPI)
       L = L + 1
       DATA(L) = TOELC
       L = L + 1
       DATA(L) = TOESC
   60  CONTINUE

       RETURN
       END

C DECK TRIM
       SUBROUTINE TRIM

*     This subroutine provides the correction of zero-speed freeboard
*     for the sinkage and trim induced by forward speeds.  Reference-
*     RICHARD C. BISHOP and NATHAN K. BALES, "A SYNTHESIS OF BOW
*     WAVE PROFILE AND CHANGE OF LEVEL DATA FOR DESTROYER-TYPE HULLS
*     WITH APPLICATION TO COMPUTING MINIMUM REQUIRED FREEBOARDS,"
*     DTNSRDC REPORT 78-SPD-811-01, JAN. 1978.  The formulae for
*     sinkage, Z0, were developed in units of feet.  Conversion
*     to meters is provided.  The formulae for trim, ang, were
*     developed in units of degrees.  Conversion to radians is made.
*     ship speed is in knots.  NBB=0 means a ship without a bow dome.

       COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
     2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
     2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25).BLEWF(25),TLEWF(25),
     2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
     2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
     2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
     2 STATNM,STATIS
       CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
       INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
     2 FBNUMB,PTNUMB,ORGOPTN
       REAL KG

       COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
       INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
       REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

       COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
       INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
       REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),TITLE(20),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

       COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
       COMPLEX II
       CHARACTER*4 PUNITS(2)
```

```
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      REAL LO
      CHARACTER*4 METER

      DATA METER /'METE'/

      CON = 1
      IF (PUNITS(1) .EQ  METER) CON = FTMETR
      DO 1 I=1,NVK

*    speed is FROUDE scaled to LO ship

      LO = 480.*CON
      VO = SQRT(LO/LPP) * VK(I)
      V2 = VO*VO
      V3 = V2*VO
      IF (NBB  EQ. 0) GO TO 20

*    ship with bow dome

      ZO = (.007848*VO + .001321*V2) * CON
      ANGO = (.015422*VO - .0021752*V2 + 5.957E-5*V3) * DEGRAD
      GO TO 30
20    CONTINUE

*    ship without bow dome

      ZO = (-.005292*VO + .001855*V2) * CON
      ANGO = (.0092648*VO - .0015692*V2 + 4.2912E-5*V3) * DEGRAD

30    CONTINUE

*    sinkage FROUDE scaled from LO ship to LPP ship.
*    sinkage and trim both defined positive.
*    freeboard correction = F - SINKAGE + FBDX*TRIM

      DO 5 J=1,NFREBD
      SNK = ZO * LPP/LO
      TRM = ANGO
      FBDZV(I,J) = FBDZ(J) - SNK + FBDX(J)*TRM
5     CONTINUE
1     CONTINUE

      RETURN
      END

C DECK TRNLAT
      SUBROUTINE TRNLAT (VCG,TL,EXCL,TLG,EXCLG)

      COMPLEX TL(3,3),EXCL(3),TLG(3,3),EXCLG(3)

      TLG(1,1) = TL(1,1)
      TLG(1,2) = TL(1,2) + VCG*TL(1,1)
      TLG(1,3) = TL(1,3)
      TLG(2,1) = TLG(1,2)
      TLG(2,2) = TL(2,2) + VCG*(TL(1,2) + TL(2,1) + VCG*TL(1,1))
      TLG(2,3) = TL(2,3) + VCG*TL(1,3)
      TLG(3,1) = TL(3,1)
      TLG(3,2) = TL(3,2) + VCG*TL(3,1)
      TLG(3,3) = TL(3,3)
      EXCLG(1) = EXCL(1)
      EXCLG(2) = EXCL(2) + VCG*EXCL(1)
      EXCLG(3) = EXCL(3)

      RETURN
      END

C DECK TWODPT
      SUBROUTINE TWODPT (KSTA,YSTA,ZSTA,NPT,PHI2D)
```

```
*     This subroutine provides two-dimensional velocity potentials for
*     oscillating cylinders of arbitrary cross section in a free surface
*     four velocity potentials associated with the individual modes
*     of oscillation, surge, sway, heave, and roll, are obtained which
*     are stored in PHI2D (frequency, offset point, mode).

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
      INTEGER      SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF GNUS,GNUF,FTMETR

      COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
      LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

      COMMON /TWOD/ YY, ZZ, ENN, ISTA
      INTEGER ISTA
      REAL YY(10,25),ZZ(10,25),ENN(4,10,25)

      COMPLEX RHS1(10), RHS2(10), RHS3(10), RHS4(10), Q1(10), Q2(10
     .        Q3(10), Q4(10), GREENV(10,10), GREENL(10,10), CTV(10,10),
     .        CTL(10,10), UV(10,10), UL(10,10), SIGIM, FAC,
     .        PHI2D(10,10,4)
      DIMENSION POTLOG(2,10,10), PTNLOG(2,10,10), CN(10), SN(10)
      DIMENSION YS(11), ZS(11), IPV(10), IPL(10)
      DIMENSION YSTA(10),ZSTA(10)
      DIMENSION SP(10),SQ(10),W1(10),W2(10)
      LOGICAL LID

      ISTA = KSTA
      FACTOR = SQRT(GRAV*LPP)
      SQRLG = SQRT(LPP/GRAV)
      DO 60 I=1,NSIGMA
      SIGMA(I) = SIGMA(I)*SQRLG
   60 CONTINUE
      DO 70 J=1,NPT
      ENN(4,J,ISTA) = ENN(4,J,ISTA)/LPP
      YS(J) = YSTA(J)/LPP
      ZS(J) = ZSTA(J)/LPP
      YY(J,ISTA) = YY(J,ISTA)/LPP
      ZZ(J,ISTA) = ZZ(J,ISTA)/LPP
   70 CONTINUE
      SQ(1) = 0.
      DO 72 N=2,NPT
      NM = N - 1
```

```
       YINT = YS(N) - YS(NM)
       ZINT = ZS(N) - ZS(NM)
       GIR = SQRT(YINT*YINT+ZINT*ZINT)
       SQ(N) = SQ(NM) + GIR
  72   CONTINUE
       NON = NPT - 1
       YINT = YY(1,ISTA) - YS(1)
       ZINT = ZZ(1,ISTA) - ZS(1)
       GIR = SQRT(YINT*YINT+ZINT*ZINT)
       SP(1) = GIR
       DO 74 N=2,NON
       NM = N - 1
       YINT = YY(N,ISTA) - YY(NM,ISTA)
       ZINT = ZZ(N,ISTA) - ZZ(NM,ISTA)
       GIR = SQRT(YINT*YINT+ZINT*ZINT)
       SP(N) = SP(NM) + GIR
  74   CONTINUE
       DO 76 N=2,NON
       NM = N - 1
       DEN = SP(N) - SP(NM)
       W1(N) = (SP(N) - SQ(N))/DEN
       W2(N) = (SQ(N) - SP(NM))/DEN
  76   CONTINUE
       DEN = SP(2) - SP(1)
       W1(1) = (SP(2) - SQ(1))/DEN
       W2(1) = (SQ(1) - SP(1))/DEN
       NM = NON - 1
       DEN = SP(NON) - SP(NM)
       W1(NPT) = (SP(NON) - SQ(NPT))/DEN
       W2(NPT) = (SQ(NPT) - SP(NM))/DEN

*  test for LID

      LID = .TRUE.
       IF (ABS(YS(NPT)) .LE. 1.0E-6) LID = .FALSE.
      NARG = NPT
      IF(.NOT.LID)  NARG = NPT-1
      NZRO = NPT + 1

*     below  two  cards are to introduce one more segment on the free
*     surface inside a cross section for removing irregular frequencies.

      YS(NZRO) = 0.
      ZS(NZRO) = 0.

      CALL GRNLOG( YS, ZS, NARG, POTLOG, PTNLOG, CN, SN)
      DO 10  K=1,NSIGMA
      SIGMA2 = SIGMA(K)**2
      SIGIM=II*SIGMA(K)
      DO 1  I=1,NON
      RHS1(I) = -ENN(1,I,ISTA)*SIGIM
      RHS2(I) = -ENN(2,I,ISTA)*SIGIM
      RHS3(I) = -ENN(3,I,ISTA)*SIGIM
      RHS4(I) = -ENN(4,I,ISTA)*SIGIM
    1 CONTINUE

*     the following four cards are to impose a rigid wall condition on
*     the waterline segment inside the section.

      IF(.NOT. LID)  GO TO 25
      RHS1(NPT) = (0.0, 0.0)
      RHS2(NPT) = (0.0, 0.0)
      RHS3(NPT) = (0.0, 0.0)
      RHS4(NPT) = (0.0, 0.0)
   25 CONTINUE

      CALL GRNFRQ( YS, ZS, NARG, SIGMA2, POTLOG, PTNLOG, CN, SN,
     .             CTV, CTL, GREENV, GREENL)

*     for the algebraic equation  AX=B, CDCOMP makes an inversion of
*     the matrix A, and CSOLVE provides the solution vector X by
*     X=(INVERTED A)B
```

```
      CALL CDCOMP( NARG, 10, CTV, UV, IPV)
      IF (IPV(NARG) .EQ. 0) GO TO 17
      CALL CSOLVE( NARG, 10, UV, RHS1, Q1, IPV)
      CALL CSOLVE( NARG, 10, UV, RHS3, Q3, IPV)
      IF (.NOT. LAT) GO TO 20
      CALL CDCOMP (NARG, 10, CTL, UL, IPL)
      IF (IPL(NARG) .EQ. 0) GO TO 17
      CALL CSOLVE (NARG, 10, UL, RHS2, Q2, IPL)
      CALL CSOLVE (NARG, 10, UL, RHS4, Q4, IPL)
   20 CONTINUE
      DO 2 I=1,NON
      PHI2D(K,I,1) =(0. , 0.)
      PHI2D(K,I,3) =(0. , 0.)
      DO 2 J=1,NARG
      FAC=GREENV(I,J)*FACTOR
      PHI2D(K,I,1) = PHI2D(K,I,1)+Q1(J)*FAC
      PHI2D(K,I,3) = PHI2D(K,I,3)+Q3(J)*FAC
    2 CONTINUE

*     PHI2DS are to be interpolated or extrapolated linearly from the
*     midpoint of the segments to the offset points.
*     QI arrays are to be used for temporary storage for PHI2DS

      DO 150 N=2,NON
      NM = N - 1
      Q1(N) = W1(N)*PHI2D(K,NM,1) + W2(N)*PHI2D(K,N,1)
      Q3(N) = W1(N)*PHI2D(K,NM,3) + W2(N)*PHI2D(K,N,3)
  150 CONTINUE
      NM = NON - 1
      Q1(1) = W1(1)*PHI2D(K,1,1) + W2(1)*PHI2D(K,2,1)
      Q3(1) = W1(1)*PHI2D(K,1,3) + W2(1)*PHI2D(K,2,3)
      Q1(NPT) = W1(NPT)*PHI2D(K,NM,1) + W2(NPT)*PHI2D(K,NON,1)
      Q3(NPT) = W1(NPT)*PHI2D(K,NM,3) + W2(NPT)*PHI2D(K,NON,3)
      DO 90 I=1,NPT
      PHI2D(K,I,1) = Q1(I)
   90 PHI2D(K,I,3) = Q3(I)
      IF(.NOT. LAT) GO TO 10
      DO 5 I=1,NON
      PHI2D(K,I,2) =(0. , 0.)
      PHI2D(K,I,4) =(0. , 0.)
      DO 5 J=1,NARG
      FAC=GREENL(I,J)*FACTOR
      PHI2D(K,I,2) = PHI2D(K,I,2)+Q2(J)*FAC
    5 PHI2D(K,I,4) = PHI2D(K,I,4)+Q4(J)*FAC
      DO 160 N=2,NON
      NM = N - 1
      Q2(N) = W1(N)*PHI2D(K,NM,2) + W2(N)*PHI2D(K,N,2)
      Q4(N) = W1(N)*PHI2D(K,NM,4) + W2(N)*PHI2D(K,N,4)
  160 CONTINUE
      NM = NON - 1
      Q2(1) = W1(1)*PHI2D(K,1,2) + W2(1)*PHI2D(K,2,2)
      Q4(1) = W1(1)*PHI2D(K,1,4) + W2(1)*PHI2D(K,2,4)
      Q2(NPT) = W1(NPT)*PHI2D(K,NM,2) + W2(NPT)*PHI2D(K,NON,2)
      Q4(NPT) = W1(NPT)*PHI2D(K,NM,4) + W2(NPT)*PHI2D(K,NON,4)
      DO 97 I=1,NPT
      PHI2D(K,I,2) = Q2(I)
   97 PHI2D(K,I,4) = Q4(I)
   10 CONTINUE
      GO TO 19
   17 WRITE (IPRIN,18) K
   18 FORMAT (//// 10X,'TWODPT -- SINGULAR MATRIX AT K=', I3)
      STOP
   19 CONTINUE

*  patch to obtain correct potential

      DO 32  K=1,NSIGMA
      DO 30  I=1,NPT
      DO 31  J=1,4
      PHI2D(K,I,J)=-CONJG(PHI2D(K,I,J))
   31 CONTINUE
```

```
      PHI2D(K,I,4)=LPP*PHI2D(K,I,4)
  30  CONTINUE
  32  CONTINUE
      DO 75 I=1,NSIGMA
      SIGMA(I) = SIGMA(I)/SQRLG
  75  CONTINUE
      DO 80 J=1,NPT
      ENN(4,J,ISTA) = ENN(4,J,ISTA)*LPP
      YY(J,ISTA) = YY(J,ISTA)*LPP
      ZZ(J,ISTA) = ZZ(J,ISTA)*LPP
  80  CONTINUE

      RETURN
      END

C DECK VELACC
      SUBROUTINE VELACC (IM,IT,GRAV,NL,NU,OMEGAE,RAO1,PHS1,RAO2,PHS2,
     2 NOMEGA,NPLANE,IPHS)

*    This routine obtains the velocity and acceleration raos and
*    phase angles for motions at the origin and at a point.
*             W.G.MEYERS, DTNSRDC, 100477

      DIMENSION OMEGAE(NOMEGA),RAO1(NOMEGA),PHS1(NOMEGA),RAO2(NOMEGA),
     2 PHS2(NOMEGA)

      GRAV2 = GRAV*GRAV
      DO 20 I=NL,NU
      OMEGE2 = OMEGAE(I)*OMEGAE(I)
      OMEGE4 = OMEGE2*OMEGE2
      DO 10 J=1,NPLANE
      IF (IT.EQ.2 .AND. J.EQ.1) RAO1(I) = RAO1(I)*OMEGE2
      IF (IT.EQ.2 .AND. J.EQ.2) RAO2(I) = RAO2(I)*OMEGE2
      IF (IT.EQ.3 .AND. J.EQ.1) RAO1(I) = RAO1(I)*OMEGE4
      IF (IT.EQ.3 .AND. J.EQ.2) RAO2(I) = RAO2(I)*OMEGE4
      IF (IT.EQ.3 .AND. IM.LT.4 .AND. J.EQ.1) RAO1(I) = RAO1(I)/GRAV2
      IF (IT.EQ.3 .AND. IM.LT.4 .AND. J.EQ.2) RAO2(I) = RAO2(I)/GRAV2
      IF (IPHS .EQ. 0) GO TO 10
      IF (IT.EQ.2 .AND. J.EQ.1) PHS1(I) = PHS1(I) + 90.
      IF (IT.EQ.2 .AND. J.EQ.2) PHS2(I) = PHS2(I) + 90.
      IF (IT.EQ.3 .AND. J.EQ.1) PHS1(I) = PHS1(I) + 180.
      IF (IT.EQ.3 .AND. J.EQ.2) PHS2(I) = PHS2(I) + 180.
  10  CONTINUE
  20  CONTINUE

      RETURN
      END

C DECK VISC
      SUBROUTINE VISC

      COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
     2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

       COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
       COMPLEX II
       CHARACTER*4 PUNITS(2)
       REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

       COMMON /RLIBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
     2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
     2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
     2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
     2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
     2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
     2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
     2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
     2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
     2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
     2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
     2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
     2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
     2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
     2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
       REAL RDBLK(2692)
       EQUIVALENCE (PSUR(1),RDBLK(1))

       DO 10 IA=1,NRANG
       DO 10 IS=1,NSIGMA
       SHPDMP(IS,IA) = 0
    10 CONTINUE
       DO 40 K=1,NSTATN
       IF (NOFSET(K) .LT. 2) GO TO 40
       CON = 4./(3.*PI)*RHO*PSUR(K)*RGB(K)**3
       DO 30 IA=1,NRANG
       DO 20 IS=1,NSIGMA
       STADMP(IS) = CON*SIGMA(IS)*RANG(IA)*EDDY(IA,K)
       STADMP(IS) = SIGMA(IS)*STADMP(IS)
       SHPDMP(IS,IA) = SHPDMP(IS,IA) + STADMP(IS)
    20 CONTINUE
    30 CONTINUE
    40 CONTINUE
       DO 50 IA=1,NRANG
       CALL SPFIT (SIGMA,SHPDMP(1,IA),HEELM(1,1,IA),NSIGMA)
       FNHE(IA) = ENCON*REVAL(HEELM(1,ISIGMA,IA),WTSI)
    50 CONTINUE

       RETURN
       END

C DECK VUNIT2
       SUBROUTINE VUNIT2 (V1, S1, V2)

*     VUNIT2 created from VUNIT ( NAVSEC-N065 ) - A M REED JULY 1976
*     unitizes plane direction vector

       DIMENSION V1(2), V2(2)

       S = SQRT( V2(1)*V2(1) + V2(2)*V2(2) )
       IF (S .LE. 0.000001*(ABS(V2(1))+ABS(V2(2)))) GO TO 2000
       S1=S
       V1(1)=V2(1)/S
       V1(2)=V2(2)/S
       GO TO 99999
2000   CONTINUE
       S1=0.0
       V1(1)=0.0
       V1(2)=0.0
99999  CONTINUE

       RETURN
```

```
      END

C DECK WAVMAK
      SUBROUTINE WAVMAK

      COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
     2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBL,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
      INTEGER LPFIDX,LRMIDX,LSVIDX
      REAL PFIDX(235),RMIDX(183),SVIDX(3)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
     2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
     1 RHOF,GNUS,GNUF,FTMETR

      COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
     2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
     2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
     2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
     2 SYCP(2) SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
     2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
     2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
     2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
     2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
     2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
     2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
     2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
     2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
     2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
     2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
      REAL RDBLK(2692)
      EQUIVALENCE (PSUR(1),RDBLK(1))

      COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
     2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
     2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
     2 LSHIPS,LTITLES
      CHARACTER*160 AS
      CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
      CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
```

```
       CHARACTER SHIPS*6,VARS*2,CYCLS*2
       INTEGER*2 OPTION

       COMMON /TELEM/ TELEM
       COMPLEX TELEM(4,9,10)

       COMPLEX T22,T24,T42,T44,T44G(10),CELM(4,9),CT44G,CDUM
       REAL IROLLG,I44G
       DATA EPS /0.25/

       FIS = SDS(1:LSDS)//'.COF'
       OPEN (UNIT=COFFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')
       READ (COFFIL) TELEM
       CLOSE (UNIT=COFFIL)

*    wavemaking (origin at VCG)

       DO 10 IS=1,NSIGMA
       JS = IS
       J = 1
       IF (IS .EQ. NSIGMA) JS = IS - 1
       IF (IS .EQ. NSIGMA) J = 3
       T44 = TELEM(J,JS,6)
       T22 = TELEM(J,JS,5)
       T24 = TELEM(J,JS,8)
       T42 = T24

*    translate to VCG

       T44G(IS) = T44 + VCG*(T24 + T42 + VCG*T22)
       SHPDMP(IS,1) = AIMAG(T44G(IS))
   10  CONTINUE
       CALL CPFIT (SIGMA,T44G,CELM,NSIGMA)

*    find natural roll frequency

       C44 = CROLL
       IROLLG = MASS*(KROLL*BEAM)**2
       I44G= IROLLG
       WPHI = SQRT(C44/I44G)
       TPHI = TPI/WPHI
       IDONE = 0
       DO 20 I=1,10
       IT = I
       TS = TPHI
       CALL CPLVAL (SIGMA,NSIGMA,CELM,WPHI,CT44G,CDUM,ISIGMA)
       A44G = REAL(CT44G)/(-WPHI**2)
       I44G= IROLLG + A44G
       IF (IDONE .EQ. 1) GO TO 30
       WPHI = SQRT(C44/I44G)
       TPHI = TPI/WPHI
       IF (ABS(TPHI-TS) .LT. EPS) IDONE = 1
   20  CONTINUE
   30  CONTINUE
       CALL FINTSP (WPHI)
       CALL SPFIT (SIGMA,SHPDMP,WMELM,NSIGMA)
       ENCON = 1./(2.*C44)
       ENWM = ENCON * REVAL(WMELM(1,ISIGMA),WTSI)

       RETURN
       END

C DECK WEDEFN
       SUBROUTINE WEDEFN (NWEVN,WEVN)

*    This routine calculates the evenly-spaced encounter wave
*    frequencies over which the response spectra are calculated.
*    The number of frequencies must be set equal to 100.
*            W.G.MEYERS, DTNSRDC, 072877

       DIMENSION WEVN(NWEVN)
       K = 0
```

```
      DWE = 0.01
      DO 110 I=1,54
      K = K + 1
110   WEVN(K) = 0.05 + (I-1)*DWE
      DWE = 0.02
      DO 120 I=1,21
      K = K + 1
120   WEVN(K) = WEVN(54)+I*DWE
      DWE = 0.10
      DO 130 I=1,10
      K = K + 1
130   WEVN(K) = WEVN(75)+I*DWE
      DWE = 0.2
      DO 140 I=1,10
      K = K + 1
140   WEVN(K) = WEVN(85)+I*DWE
      DWE = 0.4
      DO 150 I=1,5
      K = K+1
150   WEVN(K) = WEVN(95)+I*DWE

      RETURN
      END

C DECK WTPELM
      SUBROUTINE WTPELM(ISTATN, PELEM)

*     writes out spline elements for 2-d potential and forces
*           W. R. MCCREIGHT  DTNSRDC  JULY,1977

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
     1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
     2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
     1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
     2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
     2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
     2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
     2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
     2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
     4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
     5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
      INTEGER LPFIDX,LRMIDX,LSVIDX
      REAL PFIDX(235),RMIDX(183),SVIDX(3)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
     2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
     2 SPTFIL,LACFIL,LAEFIL

      COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
      LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

      DIMENSION  DATA(320)
      COMPLEX PELEM(4,9,40)

      IF (NOFSET(ISTATN) .LE. 0) RETURN
      IMMIN=1
      IF (.NOT. VRT) IMMIN=2
      IMMAX=4
      IF (.NOT. LAT) IMMAX=3
      IMDEL=2
      IF (VRT .AND. LAT) IMDEL=1
      ISGMX=NSIGMA-1
      DO 1  ISIGMA=1,ISGMX
      NEXT=1
      NNODE=NOFSET(ISTATN)
      DO 2  J=1,NNODE
      DO 3  IMODE=IMMIN,IMMAX,IMDEL
      DO 4  I=1,4
      IDX=(IMODE-1)*10+J
      DATA(NEXT)=REAL(PELEM(I,ISIGMA,IDX))
      DATA(NEXT+1)=AIMAG(PELEM(I,ISIGMA,IDX))
      NEXT=NEXT+2
    4 CONTINUE
    3 CONTINUE
    2 CONTINUE
      NDATP=NEXT-1
      INDEX=(ISIGMA-1)*NSTATN+ISTATN

*        change for VAX-11 version.
*   CDC     CALL WRITMS(POTFIL,DATA,NDATP,INDEX)

      WRITE (POTFIL,REC=INDEX) DATA

    1 CONTINUE

      RETURN
      END

C DECK XMSSC
      SUBROUTINE XMSSC (IPH,B2,MSLC,NLCH,RMSLC,RMSSC)

      DIMENSION B2(NLCH)

      REAL MSLC(24),MSSC
      MSSC = 0.
      LH = IPH - 1
      DO 10 IH=1,NLCH
      LH = LH + 1
      IF (LH .GT. 24) LH = LH - 24
      MSSC = MSSC + B2(IH)*MSLC(LH)
   10 CONTINUE
      KH = IPH + 5
      IF (KH .GT. 24) KH = KH - 24
      RMSLC = MSLC(KH)
      RMSSC = MSSC

      RETURN
      END
```

REFERENCES

1. Meyers, W. G., T. R. Applebee, and A. E. Baitis, "User's Manual for the Standard Ship Motion Program, SMP," DTNSRDC/SPD-0936-01 (Sep 1981).

2. Meyers, W. G. and A.E. Baitis, "SMP84: Improvements to Capability and Prediction Accuracy of the Stan'ard Ship Motion Program, SMP81," DTNSRDC/SPD-0936-04 (Sep 1985).

3. Meyers, W. G., C. J. Bennett, and T. R. Applebee, "User's Manual for the Simulation Time History and Access Time History Programs," CARDEROCKDIV, NSWC/SHD-1297-01 (Feb 1993).

4. Salvesen, N., E. O. Tuck, and O. Faltinsen, "Ship Motions and Sea Loads," In *Trans. SNAME*, Vol. 78 (Nov 1970).

5. Norton Editor User's Manual Symantec Corp., copyright 1993.

6. HALO Professional Graphics Kernel System Version 2.0 Media Cybernetics Inc., copyright 1990,1991.

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency Under the Jurisdiction of the Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON **Technical Director**	**MICHAEL D. SHALER** **COL, AR** **Commanding**

Research accomplished under contract
for the Department of the Army

Anacapa Sciences, Inc.

Technical review by

John E. Stewart
Major Robert D. Watkins
Dennis C. Wightman

NOTICES

~~DISTRIBUTION: Primary distribution of this report has been made by ARI. Please address correspondence concerning distribution of reports to: U.S. Army Research Institute for the Behavioral and Social Sciences, ATTN: PERI-POX, 5001 Eisenhower Ave., Alexandria, Virginia 22333-5600.~~

FINAL DISPOSITION: This report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS
--

2a. SECURITY CLASSIFICATION AUTHORITY
--

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
--

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution is unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
ASI690-341-91

5. MONITORING ORGANIZATION REPORT NUMBER(S)
ARI Research Report 1604

6a. NAME OF PERFORMING ORGANIZATION
Anacapa Sciences, Inc.

6b. OFFICE SYMBOL *(If applicable)*
--

7a. NAME OF MONITORING ORGANIZATION
U.S. Army Research Institute
Aviation R&D Activity

6c. ADDRESS *(City, State, and ZIP Code)*
P.O. Box 489
Fort Rucker, AL 36362-5000

7b. ADDRESS *(City, State, and ZIP Code)*
ATTN: PERI-IR
Fort Rucker, AL 36362-5354

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
U.S. Army Research Institute for the Behavioral and Social Sciences

8b. OFFICE SYMBOL *(If applicable)*
PERI-I

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
MDA903-87-C-0523

8c. ADDRESS *(City, State, and ZIP Code)*
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
63007A	795	3405	C06

11. TITLE *(Include Security Classification)*
Training Effectiveness of the AH-64A Combat Mission Simulator for Sustaining Gunnery Skills

12. PERSONAL AUTHOR(S)
Hamilton, David B.

13a. TYPE OF REPORT
Interim

13b. TIME COVERED
FROM 88/07 TO 91/03

14. DATE OF REPORT *(Year, Month, Day)*
1991, November

15. PAGE COUNT

16. SUPPLEMENTARY NOTATION
All research on this project was technically monitored by Charles A. Gainer, Chief, U.S. Army Research Institute Aviation Research and Development Activity (ARIARDA), Fort Rucker, AL

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
05	08	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Transfer of training AH-64A helicopter
Training effectiveness Aircrew training
Skill sustainment (Continued)

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This research evaluated the effectiveness of the AH-64A Combat Mission Simulator (CMS) for sustaining crew gunnery proficiency. Although the research was originally designed to be conducted over a 12-month period, the time period was shortened to 6 months to overcome problems with crew attrition and to meet project deadlines. Baseline gunnery performance was measured both on the live-fire gunnery range and in the CMS for 30 AH-64A crews from an operational cavalry brigade. Subsequently, the crews were divided into two groups. The control group continued normal unit training but was restricted from gunnery practice in the CMS. Each crew in the simulator group received five scenario-based gunnery training sessions in the CMS and normal unit training but was restricted from live-fire practice in the aircraft. Six months after the baseline measures, crew gunnery performance was again evaluated on the live-fire gunnery range and in the CMS for the 18 crews that remained in the experiment. The results failed to indicate CMS gunnery training effectiveness: Gunnery skill enhancement was not detected in the simulator group's performance and gunnery skill

(Continued)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Charles A. Gainer

22b. TELEPHONE *(Include Area Code)*
(205) 255-4404

22c. OFFICE SYMBOL
PERI-IR

Previous editions are obsolete.

ARI Research Report 1604

18. SUBJECT TERMS (Continued)

Gunnery training
AH-64A combat mission simulator
Crew gunnery qualification

19. ABSTRACT (Continued)

decay was not found in the control group. The failure to demonstrate the training effectiveness of the CMS is probably due to the high initial skill levels of the aviators and the lack of skill decay in the control group over a 6-month period.

Research Report 1604

Training Effectiveness of the AH-64A Combat Mission Simulator for Sustaining Gunnery Skills

David B. Hamilton
Anacapa Sciences, Inc.

ARI Aviation R&D Activity
Charles A. Gainer, Chief

Training Research Laboratory
Jack H. Hiller, Director

U.S. Army Research Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

Office, Deputy Chief of Staff for Personnel
Department of the Army

November 1991

Army Project Number 2Q263007A795	**Training Simulation**

FOREWORD

This research was performed within the Training Research Laboratory by the U.S. Army Research Institute Aviation Research and Development Activity (ARIARDA), Fort Rucker, Alabama, and was sponsored by the Standards in Training Commission (STRAC). The research was conducted in response to two taskings: One from the U.S. Army Aviation Center (USAAVNC) and one from the Department of the Army. It was accomplished as an annex to the Memorandum of Agreement between ARIARDA and the Directorate of Training and Doctrine, dated 15 March 1984.

Over the past two decades, the Army has made a significant investment in rotary-wing aviator training with the development and acquisition of motion-based visual flight simulators. One example of this type of simulator is the AH-64A Combat Mission Simulator (CMS). With the high expense of aircraft operations and the decreased availability of live munitions, AH-64A gunnery training in the CMS has been viewed as a safe, cost-effective alternative to aircraft training.

High-fidelity flight and weapons simulators have been deployed to support aircrew training in operational aviation units. However, little empirical data exist to document the training effectiveness of the simulators. To support the Army deployment of the CMS, a research approach was designed to generate empirical data on the effectiveness of the AH-64A CMS for sustaining gunnery skills. The research was designed to test the effectiveness of simulator gunnery training in live-fire gunnery exercises. This document reports the results of that research.

This report will serve as a source of information about the training effectiveness and capabilities of the AH-64A CMS. Results were briefed to representatives of STRAC in December 1990 and USAAVNC in January 1991. Other briefings to operational personnel were conducted from January through March 1991. The information in this report was used to rewrite the Gunnery Manual TC 1-140 and will be effective for developing simulator training strategies for aerial gunnery.

EDGAR M. JOHNSON
Technical Director

ACKNOWLEDGMENTS

The author wishes to express appreciation to a number of persons for their contributions to this research.

Several individuals from ARIARDA contributed to this project. Charles A. Gainer, Chief, ARIARDA, Fort Rucker, Alabama, served as the Contracting Officer's Technical Representative. Dennis Wightman (Technical Team Leader, Simulation Research Program) and Captain Dale Weiler (Research and Development Coordinator) provided administrative assistance. Joan Blackwell (Research Psychologist) provided advice and participated in the initial research effort of the project. Larry Murdock (Data Processing Manager) was responsible for running the initial statistical analyses.

Members of the Anacapa Sciences, Inc., staff also contributed to this project. Most notably, Catherine R. Kjellsen (Aviation Research Technician) was invaluable in the on-site data collection effort. D. Michael McAnulty consulted on the design, analyses, and presentation of the data.

The leadership and support of Colonel Thomas J. Konitzer (Commander, 6th Cavalry Brigade, Air Combat), made this research project possible. Although they cannot all be acknowledged individually, the author thanks the other unit commanders, operations officers, instructor pilots, and aviators who provided the leadership, coordination, instruction, and participation necessary to complete the research.

TRAINING EFFECTIVENESS OF THE AH-64A COMBAT MISSION SIMULATOR FOR SUSTAINING GUNNERY SKILLS

EXECUTIVE SUMMARY

This report describes the methods and results of an experiment designed to measure the effectiveness of the AH-64A Combat Mission Simulator (CMS) for sustaining gunnery skills in Army aviators. The research was conducted by the U.S. Army Research Institute Aviation Research and Development Activity.

Requirement:

The Army has made a significant investment in the development and acquisition of motion-based, visual flight and weapons simulators for training rotary-wing aviators. Most of the simulators have been deployed to operational units to help reduce the training cost of sustaining flight and gunnery skills in proficient aviators. However, the effectiveness of flight simulators in augmenting unit gunnery training has not been demonstrated. Empirical data are required to demonstrate that flight simulators are effective in sustaining gunnery skills and to determine the extent that simulator training can be used to conserve resources such as aircraft flight time and live ammunition.

The research objectives of this experiment were (a) to determine the effectiveness of the CMS for sustaining crew gunnery skills and (b) to provide information on the optimum combination of aircraft and CMS training for sustaining those skills.

Procedure:

An operational cavalry unit participated in a forward transfer-of-training experiment designed to meet the research objectives. An initial evaluation of AH-64A crew gunnery performance was conducted both during a live-fire exercise and during a CMS test scenario. Subsequently, crews were assigned to one of two groups. The simulator group crews continued normal unit training and received scenario-based CMS gunnery training but were restricted from live-fire training. The control group crews received the normal unit training but were restricted from CMS gunnery training. The training phase of the research, originally scheduled for a year, was shorted to 6 months to meet project schedules and to minimize crew attrition. Crew gunnery performance was measured again during a final live-fire exercise and in the CMS.

Findings:

Analysis of the initial and final performance tests in the CMS showed that after five gunnery training sessions in the CMS performance was consistently but not significantly improved in the experimental group. However, the skill improvement did not transfer to the live-fire range. The simulator group's performance was not significantly better than the control group crew's performance during the final live-fire exercise. In addition, neither group showed any indication of gunnery skill decay over the course of the experiment. Because the results did not demonstrate the effectiveness of the CMS for sustaining gunnery skills over 6 months, no conclusion can be drawn about the optimum combination of CMS and aircraft training.

Utilization of Findings:

The costs of AH-64A gunnery training resources (e.g., flight and range time, ammunition) have increased the Army's dependence on flight simulators for training that was previously accomplished in the aircraft. However, the Army has not had empirical data about the training effectiveness of the CMS for sustaining gunnery skills to determine the optimal utilization of the flight simulator. Although the data are limited by the relatively short experimental period, two recommendations are presented on the basis of the research. First, if aircraft hours and other forms of gunnery training continue at the levels observed in this research, CMS gunnery training may be required only on a semiannual or quarterly basis. If the support for aircraft hours and other gunnery training is reduced, gunnery skills may decay in less than 6 months and additional CMS training will be required to maintain gunnery skills. Second, further research is required to investigate gunnery skill decay in proficient aviators over a 12- to 18-month period.

TRAINING EFFECTIVENESS OF THE AH-64A COMBAT MISSION SIMULATOR FOR SUSTAINING GUNNERY SKILLS

CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

TRAINING EFFECTIVENESS OF THE AH-64A COMBAT MISSION SIMULATOR FOR SUSTAINING GUNNERY SKILLS

Introduction

During the past two decades, the U.S. Army has committed hundreds of millions of dollars to the development and acquisition of motion-based visual flight simulators to augment helicopter pilot training. The simulator cockpits are constructed from the same components used to build the aircraft and consequently produce high-fidelity simulations of the controls and displays in the aircraft. The hardware environments are supported with powerful mainframe computer systems capable of generating and displaying the results of aircraft, aerodynamic, meteorological, geographic, tactical, and weapons modeling.

Flight simulators are a means of obtaining operational readiness at an acceptable cost. Relatively inexpensive simulator training is used as a cost-effective alternative to more expensive aircraft training. In fact, the primary justification for the Army's Synthetic Flight Training System (SFTS) has been the economy of simulator-for-aircraft substitution (see Hopkins, 1979).

There are at least two other benefits of simulator-based training. One is increased safety. A large number of emergency procedures that are inherently dangerous in the aircraft can be practiced in the simulator (e.g., engine or tail-rotor failures). Aviator proficiency in these procedures translates into saved lives and equipment. Day-to-day aircraft operations are not likely to provide the practice in these maneuvers that simulators can.

The second major benefit of simulators is that scenarios can be created that model the danger and complexity of the modern battlefield. A realistic force-on-force training scenario is difficult (or impossible) to accomplish in the aircraft during peacetime. By necessity, training at Army gunnery ranges arrays a maximum of firepower against only the semblance of a threat and consists of regimented procedures designed to maximize the safety of the participants and the surrounding community. In contrast, Army tacticians foresee the modern battlefield as dynamic and dangerous. With an interactive threat, unlimited ammunition, and unrestricted firing opportunities, flight simulators can potentially train Army aviators to fight and survive in a realistic wartime environment.

The Army has acquired 39 high fidelity flight simulators to support aviator training for the AH-1 Cobra, UH-60 Black Hawk, CH-47 Chinook, and AH-64 Apache aircraft. The majority, including 7 AH-1 Flight and Weapons Simulators (AH1FWSs), 15 UH-60 Flight Simulators (UH60FSs), 5 CH-47 Flight Simulators (CH47FSs), and 5 AH-64A Combat Mission Simulators (CMSs), have been delivered to operational aviation units for unit training. The remainder, consisting of 2 AH1FWSs, 2 UH60FSs, 1 CH47FS, and 1 CMS, are used for institutional training at the U.S. Army Aviation Center (USAAVNC).

With the acquisition of these resources, the Army has committed simulators to accomplish two different types of training: institutional and unit. Institutional training refers to the initial flight and weapon systems training given to Army aviators. Unit training refers to the training given to Army aviators after they have completed institutional training and have been assigned to an operational unit. The primary goal of institutional training is the acquisition of individual skills. In contrast, the primary goal of unit training is the acquisition of crew and team skills and the sustainment of all skills (i.e., individual, crew, and team).

With the acquisition of the simulators, the Army initiated research to address questions about the effectiveness of the rotary wing simulators and about the tasks that can be trained in the simulators. Previous research had demonstrated the value of simulators for the acquisition of basic flight and procedural skills in fixed wing aircraft (see Jacobs, Prince, Hays, & Salas, 1990, and Valverde, 1973, for reviews). However, the number of experiments conducted on rotary wing simulators was small by comparison (Holman, 1979; Bridgers, Bickley, & Maxwell, 1980; Luckey, Bickley, Maxwell, & Cirone, 1982). Unfortunately, the experiments that demonstrated the effectiveness of existing simulators had not also identified the characteristics of the simulators that mediate the effective transfer of skills (Orlansky & String, 1977). Without a clear understanding of the mechanisms of successful skill acquisition in fixed wing simulators, the Army could not assume that the fixed wing results would generalize to rotary wing simulators.

Another theoretical and practical question of concern to the Army is whether skills that can be acquired in the simulator can also be sustained in the simulator. The effectiveness of simulators has not been as thoroughly researched for skill sustainment as for skill acquisition. In the study of skill sustainment, the proficient aviator can be assumed to have learned the environmental stimuli that determine the appropriate actions and reactions in the aircraft. However, once skills are refined in the aircraft, the simulator may not provide the necessary stimuli to maintain the skill. Thus, without specific knowledge about the mechanisms of successful transfer-of-training, questions of the effectiveness of a particular simulator for the acquisition or sustainment of skills must be answered empirically.

Background

The research described in this report was initiated as a result of three administrative events, which are described in the following three sections.

Flight simulation plan audits. Almost all the resources expended by the Army on the SFTS program have been for the development and acquisition of the simulators. The resources devoted to research on how to use the simulators effectively have been small by comparison. Thus, the specific effects that flight simulators are capable of accomplishing in Army aviator training have not been empirically determined.

In two audits of the SFTS, first in 1981 and again in 1984, the Army Audit Agency (AAA) recognized the lack of research documenting the effectiveness of simulators for sustaining helicopter flight and gunnery skills. The AAA reports (U.S. Army Audit Agency, 1982, 1985) stated that, although flight simulators had reduced the training costs and improved training at the USAAVNC, the Army had not determined the effects that flight simulators have on unit training. Specifically, both reports admonished the Army for the operational tests conducted on the SFTS and concluded that the Army had not adequately quantified the return on its investment in flight simulators procured for unit training.

DA tasking. In 1986, the Department of the Army (DA) tasked the Army Research Institute Aviation Research and Development Activity (ARIARDA), through the Training and Doctrine Command (TRADOC), to plan and initiate postfielding training effectiveness analyses (TEAs) of each of the Army's flight simulator systems. The TEAs were intended to investigate the utilization and training effectiveness of Army flight simulator systems in operational field units and to provide a basis for developing effective unit training strategies. In respons^ to the tasking, ARIARDA developed a research plan comprising a series of related research projects (U.S. Army Res arch Institute Aviation Research and

Development Activity, 1986; Cross & Gainer, 1987). Each project was designed to investigate the effectiveness of a flight simulator system for training a set of specific tasks (e.g., contact and emergency flight tasks, weapons tasks) in an operational environment. Four of the projects have subsequently been completed with the AH1FWS (Kaempf, Cross, & Blackwell, 1989; Kaempf & Blackwell,1990; McAnulty & Kaempf, 1991).

Gunnery manual revisions. Concurrent with the DA tasking, the Department of Tactics and Simulation (DOTS; formerly the Department of Gunnery and Flight Systems) proposed revisions to the helicopter gunnery training manual (FM 1-140; Department of the Army, 1986). FM 1-140 defines the training requirements and performance standards for the Army's aerial gunnery training program. In response to increasing pressure to reduce the requirements for training ammunition, DOTS proposed significant changes to the crew gunnery training requirements and standards for the AH-64A aircraft in the coordinating draft of the revised helicopter gunnery manual (TC 1-140; USAAVNC, 1988). For example, DOTS proposed to conduct all AH-64A crew gunnery training and qualification in the CMS. No ammunition was provided for crew training and qualification; ammunition was provided only for training attack helicopter teams and conducting combined arms live-fire and joint air attack team (JAAT) exercises. While considering the substitution of simulator gunnery training for live-fire gunnery training, DOTS personnel identified a need for information on the effectiveness of the CMS for gunnery training.

Twenty-two months later, DOTS released the approved draft of the helicopter gunnery manual (TC 1-140; USAAVNC, 1990). In this version of TC 1-140, the proposal that all AH-64A crew gunnery be conducted in the CMS was dropped and the available training ammunition was redistributed among the gunnery tables, this time with more for the crew tables and less for the team tables. The document continued to predict that "reductions in service ammunition for training are inevitable" and suggested that unit commanders use the CMS and AH-1 simulator to "help aircrews maintain their proficiency between live-fire exercises and reduce the need to use live ammunition for certain tasks" (p. B-1).

CMS Effectiveness for Sustaining Gunnery Skills

Operational unit commanders are faced with increasing pressure to reduce training ammunition requirements and use the most efficient and effective mix of simulator and aircraft training. There is little empirical data to demonstrate the effectiveness of flight simulators in augmenting unit gunnery training. Empirical data are required to demonstrate that flight simulators can effectively train gunnery skills and to determine the extent that training conducted in simulators can be used to conserve training resources such as aircraft flight time and live ammunition.

This report describes research on the training effectiveness of the AH-64A CMS for sustaining gunnery skills. It is one of a group of projects planned by ARIARDA in response to the DA tasking for TEAs on each of the Army's simulators. In addition, ARIARDA agreed to focus the initial TEAs on the effectiveness of the CMS for training and sustaining crew gunnery skills at the request of the Army Standards in Training Commission (STRAC) and DOTS. Therefore, the research was designed to meet two major objectives:

- determine the effectiveness of the CMS for the sustainment of crew gunnery skills, and
- provide data to establish an optimum combination of aircraft and flight simulator training for the sustainment of crew gunnery skills.

In addition to the objectives described above, STRAC and DOTS requested an evaluation of the ammunition requirements and gunnery standards for AH-64A crew qualification published in the revised helicopter gunnery manual. The research addressing these issues is published in a separate report (Hamilton, 1991).

Design Considerations

The value of any training experience depends upon how effectively training transfers to the operational task. In the case of flight simulators, the amount of aircraft training that can be conserved as a function of simulator training is a direct measure of the training effectiveness of the simulator. The transfer of skills, facts, and attitudes can be positive or negative. Positive transfer occurs when learning simulator skills facilitates the acquisition of aircraft skills. Negative transfer occurs when learning simulator skills interferes with the acquisition of aircraft skills.

The methods for quantifying the transfer of training and training effectiveness of aircraft simulators are well developed and quantitative (Roscoe & Williges, 1980; Roscoe, 1971), especially for skill acquisition. Basically, the method uses a simple ratio to quantify the value of training time in the simulator in terms of the aircraft time saved. At a minimum, some measurable difference must exist between the performance of the experimental and control groups to demonstrate training effectiveness. If the information obtained from training research is sufficiently detailed, the ratio can be calculated for incremental amounts of time in the simulator to describe an entire function called the incremental transfer effectiveness function. The function is described as being negatively decelerated, meaning that the effectiveness of any training experience decreases with exposure to that experience. The hypothetical shape of the function is demonstrated by the curve labeled "training effectiveness" in Figure 1.

The design of research that demonstrates skill sustainment is different from research that demonstrates skill acquisition. The differences are illustrated by the learning curve labeled "skill" in Figure 1, which demonstrates how skills are typically acquired. Initially, with no skill level present, training is highly effective in increasing skill levels. As skill is acquired, increasing amounts of training produce less skill acquisition and, at some point, becomes skill sustainment. Research to quantify skill acquisition assumes that both the experimental and control groups are on the initial, accelerating part of the curve with low skill levels and that training effectiveness can be demonstrated as soon as the simulator is effective in transferring skills to the experimental group.

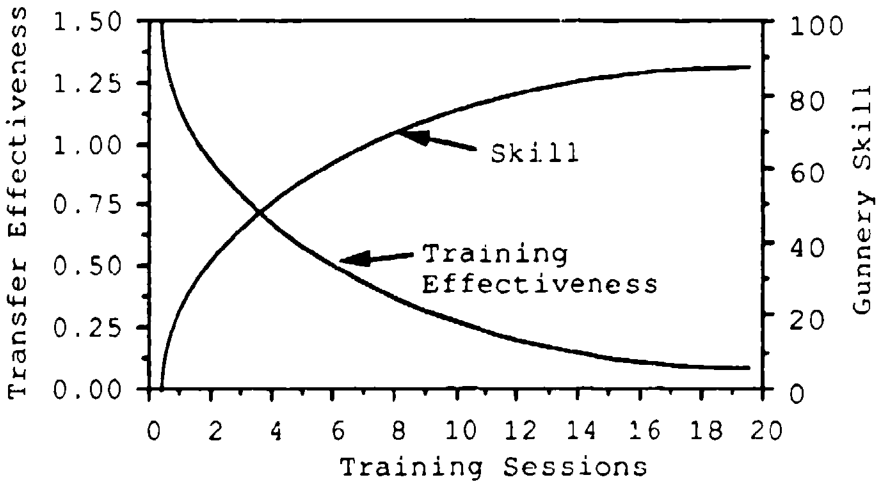


Figure 1. Hypothetical relationships between the acquisition of skill and training effectiveness.

In contrast, research to quantify skill sustainment assumes that both the experimental and control groups are on the asymptotic part of the curve. Training effectiveness is difficult to demonstrate by transferring skills to the experimental group when both groups already have high levels of skill, simply because very little further learning can occur. If skills are sufficiently well developed before the initiation of the research, the only way to bring about the difference in performance needed to demonstrate training effectiveness is to allow the control group's skills to decay. If the simulator is effective in maintaining the experimental group skills while the control group skills decay, then training effectiveness is demonstrated. If the simulator is not effective in sustaining the experimental group skills, they will decay along with the control group.

Thus, the question of how long it takes for AH-64A gunnery skills to decay is critical to the design of this research project. Ruffner and Bickley (1983) and Ruffner, Wick, and Bickley (1984) studied the decay of procedural and psychomotor flight skills in active duty and reserve Army aviators. Ruffner et al. stated that skill decay may have a critical period between 6 and 12 months. Before this period, little proficiency loss is expected; after the period, operationally important loss occurs, followed by a very long period where additional loss is relatively small.

Initial Research Effort

The research described in this report was preceded by an unsuccessful attempt to conduct a CMS TEA project. The initial research design proposed that AH-64A crew gunnery skills be measured during a pretest live-fire gunnery exercise. Subsequently, each crew would be assigned to one of three different training groups: a control group and two experimental groups. All groups would receive the normal program of instruction for the unit. One experimental group would receive CMS gunnery training; the other group would receive dry-fire gunnery training in the aircraft; and the control group would be restricted from gunnery training in either the CMS or in the aircraft. The gunnery training would be controlled in each group for 1 year. At that time, crew gunnery skills would again be evaluated during a posttest live-fire exercise. The effectiveness of the CMS would be evaluated by comparing the differential performance of the three groups between the pretest and posttest exercises.

The research was begun as described above when live-fire performance data were collected on 15 crews. The Army unit participating in the research was unable to assign other crews to the project because of anticipated personnel turnover. Consequently, live-fire data were collected 3 months later for an additional 12 crews. By that time, 4 of the original crews were unable to participate in the research because at least one of the crewmembers was assigned to another unit. At the initiation of the training phase of the research, there were 9 crews in the control group, 8 crews in the aircraft training group, and 6 crews in the simulator training group.

Within 1 month, crew attrition was so high that the research design was reevaluated. Several factors contributed to the attrition of crews. A major storm damaged many of the operational aircraft at the participating installation. Because of the lack of aircraft, some aviators were transferred to other units or types of aircraft. In addition, some crewmembers were transferred to another unit because of a high priority training mission. Finally, crew attrition was exacerbated because the loss of either crewmember constituted the loss of the entire crew. The possibility of conducting the research over the course of an entire year was eventually precluded by the attrition of participating crews. Therefore, an alternative research plan was developed and the current research effort was initiated.

Method

General Procedures

The revised research plan was divided into three phases (see Figure 2). During Phase 1, an initial evaluation of AH-64A crew gunnery performance was conducted during a live-fire exercise and during a CMS test scenario. During the live-fire exercise, the crew fired a set of crew gunnery engagements developed by the participating unit and referred to as Table VIII. During the CMS test, the crews fired against targets designated in a mission scenario developed by the researchers and the unit standardization instructor pilots (SIPs). The primary measures of gunnery performance collected during the live-fire exercises and the CMS test scenario were target effect and engagement time. In addition, the participating aviators completed a demographic survey describing their skill and training at the initiation of the research.

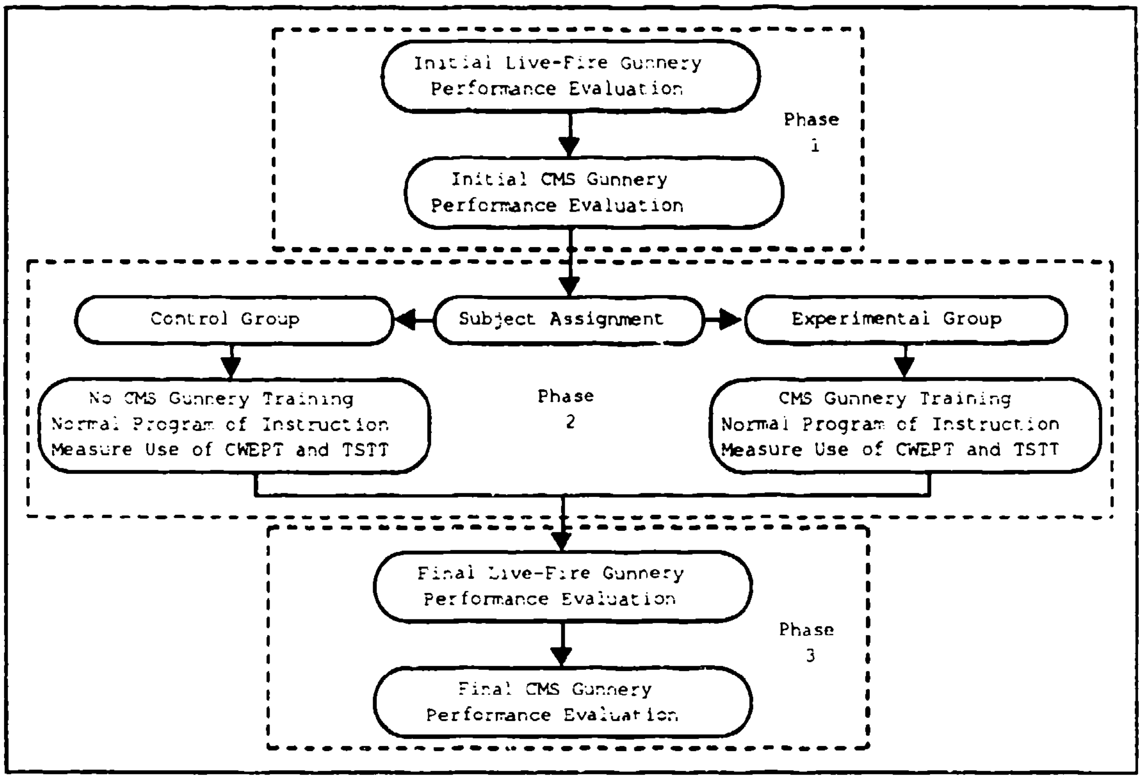


Figure 2. Flow chart of principal phases in the experimental design.

In Phase 2, crews were assigned to one of two groups: an experimental group that received scenario-based gunnery training in the CMS and a control group that was restricted from gunnery training in the CMS. The training phase of the research was shortened to only 6 months to achieve project schedules and to minimize crew attrition. The frequency of other non-CMS gunnery training activities was also recorded during this period.

In Phase 3, crew gunnery performance was measured during a final live-fire exercise and in the CMS. The effectiveness of the CMS was evaluated by measuring the differential performance of the training groups between the pretest and posttest in the CMS and during the live-fire exercises.

Apparatus

Two flight systems (the AH-64A aircraft and the AH-64A CMS) and a scoring system were used during this research. Each of these systems is described in the following sections.

AH-64A aircraft. The AH-64A (see Figure 3) is a twin engine, four-bladed helicopter with a maximum gross weight of 17,650 pounds and an approximate height, width, and length (excluding the rotor system) of 15 ft, 17 ft, and 49 ft, respectively. The two crewmembers, a pilot (PLT) and a copilot/gunner (CPG), are seated in tandem with the PLT behind and above the CPG. The AH-64A is a weapons platform equipped with point target (Hellfire missile), area weapon (30 mm chain gun), and aerial rocket (2.75-inch folding-fin type) systems. The helicopter is equipped with a laser range finder/designator (LRF/D), a pilot night vision system (PNVS), and a CPG target acquisition and designation system (TADS) that allow the crew to operate the helicopter at night and under adverse weather conditions. The AH-64A can acquire and fire on targets in a large number of different operating modes. Additionally, an on-board video recorder subsystem (VRS) can record the imagery and symbology being displayed by either the PNVS or TADS. The operation of the aircraft is described in the Operator's Manual for the AH-64A Helicopter (Department of the Army, 1984).

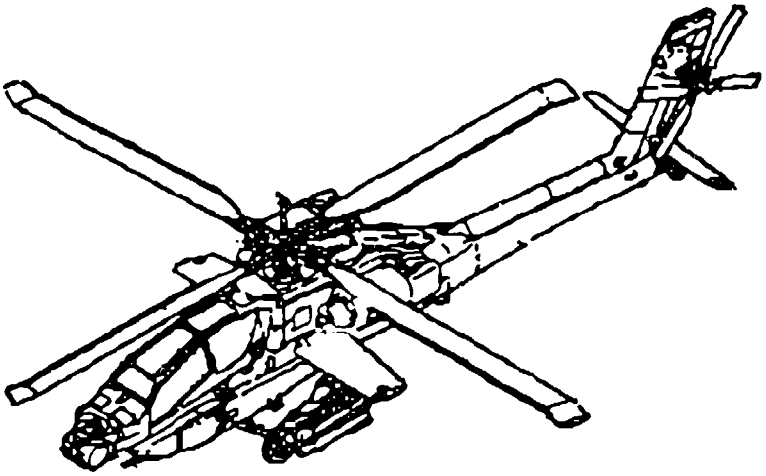


Figure 3. Diagram of the AH-64A aircraft.

AH-64A CMS. The evaluation of the gunnery training effectiveness of the AH-64A CMS was the primary focus of this research. The AH-64A CMS is a flight and weapons simulator designed for training aviators in the use of the AH-64A Apache helicopter. The CMS consists of two flight simulator compartments (PLT and CPG), each having a six-degree-of-freedom motion base. Each compartment simulates the helicopter environment using a multichannel digital image generator, three pairs of loudspeakers, a subwoofer, and a seat vibrator. The simulator is operated in an integrated mode for crew training or in an independent model for individual training. Additionally, each compartment has an instructor/operator (I/O) station and an observer station. The operation and capabilities of the CMS are fully described in the Operator's Manual for the AH-64A (Apache) Combat Mission Simulator (Department of the Army, 1988).

Area Weapons Scoring System. The Army has sponsored the development of a scoring system for attack helicopter live-fire training and evaluation designated the Area Weapons Scoring System (AWSS). The AWSS was used during the initial and final live-fire exercises for objective scoring of AH-64A gunnery performance. Although the Army plans to acquire a number of the systems, the AWSS used in this research was the proof-of-principle system installed on the Dalton-Henson Multipurpose Range Complex at Fort Hood, Texas.

The AWSS consists of the Ballistic Scoring Subsystem (BSS) for 30 mm projectiles, the Detonation Scoring Subsystem (DSS) for rockets, and the Computer Scoring Subsystem (CSS) for score calculation, display, and hard-copy production. The BSS (see Figure 4) uses special purpose, Doppler radar sensors to detect the rounds that penetrate a 15 m radius fan in front of each target. The 30 mm rounds that penetrate the Doppler fan are counted as hits; those outside the fan are counted as misses. No information about the exact location of the hits or misses is provided by the BSS, but AWSS personnel could detect when the target was struck by a burst.

The DSS (see Figure 5) is an acoustical system that determines the geographic location of rocket impacts. It consists of 10 microphone sensors placed within 1000 m of the target. During a rocket engagement, each sensor transmits the acoustical signal that it receives to the CSS. Using the known position of the sensors and the physics of sound propagation, the CSS analyzes the signals from several sensors to compute the impact point, cross range miss distance, and down range miss distance for each rocket. The system reliably determines the location of rocket impacts up to approximately 350 m from the target. Rockets falling

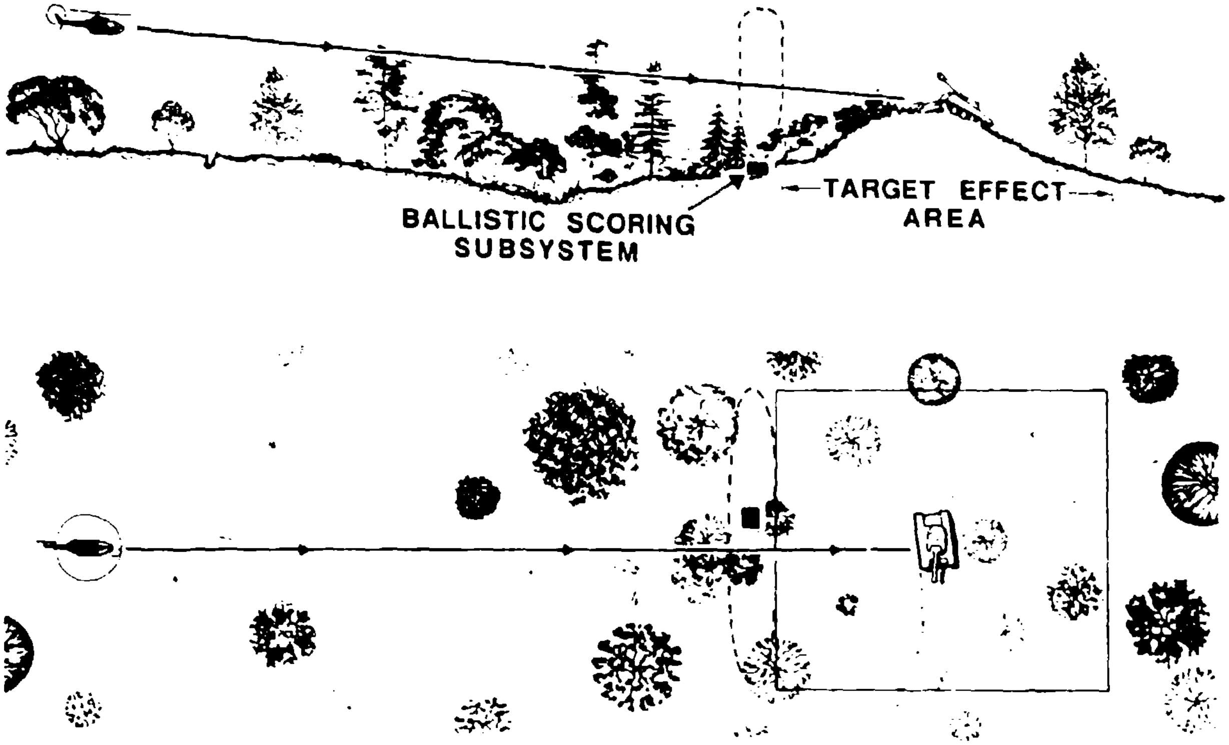


Figure 4. Diagram of the Ballistic Scoring System of the Area Weapons Scoring System.

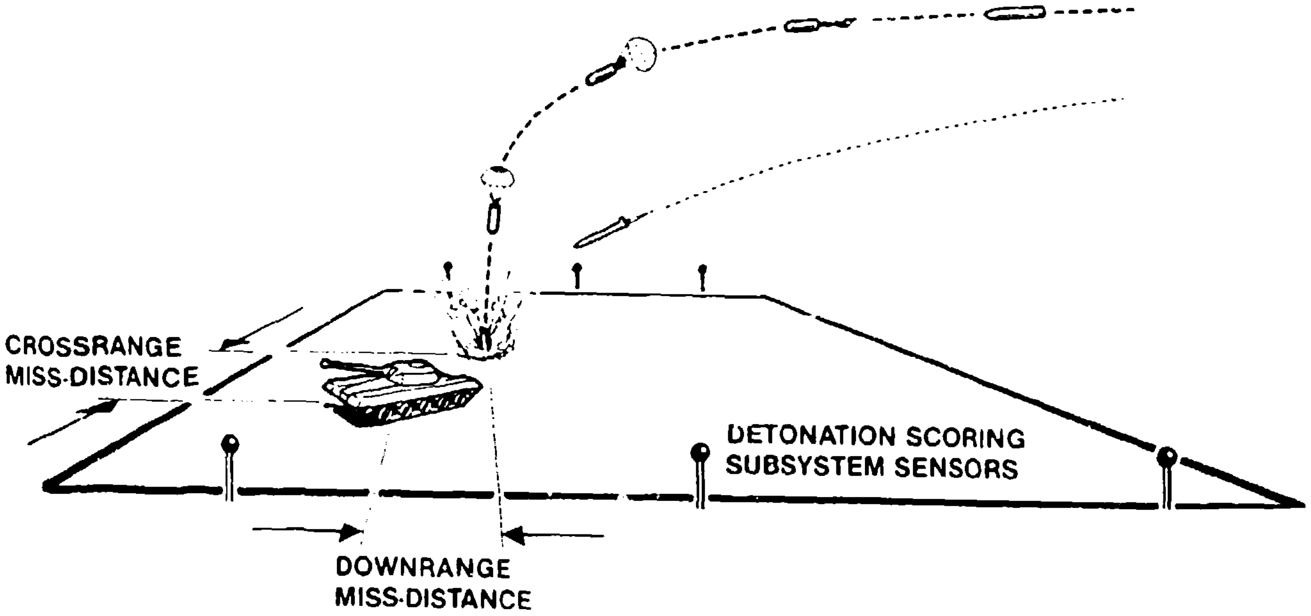


Figure 5. Diagram of the Detonation Scoring System of the Area Weapons Scoring System.

beyond the range of 350 m are not detected or have a large location errors.

The proof-of-principle AWSS had three notable limitations associated with the DSS. First, at the initiation of the project, the system was not reliably scoring Multi-Purpose Submunition (MPSM) rocket engagements. MPSM rockets were not used during the live-fire exercises. Second, the system was not reliably scoring multiple rocket engagements. Third, the acoustically based DSS was susceptible to interference from any other loud events such as the 30 mm gun firing. Because of these limitations, the four rockets that made up each engagement were fired individually with approximately 30 to 60 seconds between launches, and no engagements were fired simultaneously.

Materials

Data forms. Two types of data forms were developed and used to collect information from participating aviators: an AH-64 CMS Gunnery Research Program Demographic Survey and a Postflight Debriefing form. The AH-64 Demographic Survey (see Appendix A) was designed to collect personal, training, flight, and gunnery range experience that was used to characterize the experience of the aviators who participated in the research. As noted in the general procedures, the survey was completed by all aviators during the initial live-fire exercises.

The Postflight Debriefing form (see Appendix B) was designed to collect information about the specific gunnery tasks performed during the training phase of the research. Each aviator was instructed to complete the form after each flight in the AH-64A aircraft, the CMS, or Cockpit, Weapons, and Emergency Procedures Trainer (CWEPT).

Live-fire crew gunnery table. The unit crew qualification table used in the experiment was designed for the Dalton-Henson Multipurpose Range Complex (see Table 1). The table contains 2 calibration and 18 normal engagements employing all three AH-64A weapon systems. It was used for both day and night training. The engagements were fired from seven firing points toward 13 targets (see Figure 6). The distance from the firing points to the targets ranged from 975 m to 2575 m for the 30 mm gun, from 3450 m to 4500 m for the rockets, and from 2100 m to 4620 m for the missiles. All engagements were fired from a stationary hover with the exception of the two 30 mm engagements that were fired from a

Table 1

Initial and Final Live-Fire Gunnery Table

Firing Point	Weapon System	Target Number	Target Distance	Rounds
1[a]	30 mm	1-7A,B	975	20
	Rockets	R3	3700	4
1	30 mm	1-7A,B	975	20
	Rockets	R3	3700	4
	Hellfire	R4	3835	1
2	30 mm	9A	1066	20
	Rockets	R2	3450	4
	Hellfire	R2	3450	2
3	30 mm	1-8A,B	1700	20
	Rockets	R3	4500	4
	Hellfire	R2	4350	1
4	30 mm	8A	1645	20
	Rockets	R2	4400	4
	Hellfire	R3	4620	1
5	30 mm	8B	1400	20
6	30 mm	9B	1100	20
7	Hellfire	43	3775	1
	Hellfire	32	2350	1
	Hellfire	31	2100	1
	30 mm	34	2575	20

Note. The 30 mm engagements employed target practice (TP) rounds and the rocket engagements employed target practice point detonating (TP/PD) warheads with Mark 66 motors; the Hellfire engagements were simulated.

[a] calibration

moving hover at firing points 5 and 6. The arrows in Figure 6 indicate the direction of movement of the targets and aircraft, if any occurred.

CMS scenario. A single gunnery scenario was developed to test and train crew gunnery performance in the simulator. The I/O situation and target handover sheet used to implement the scenario are presented in Appendix C. The scenario exercised all weapons systems (30 mm, rockets, and missiles), target modes (moving and stationary), and aircraft modes (stationary and moving hover) at a variety of target ranges. The scenario contains engagements similar to those in

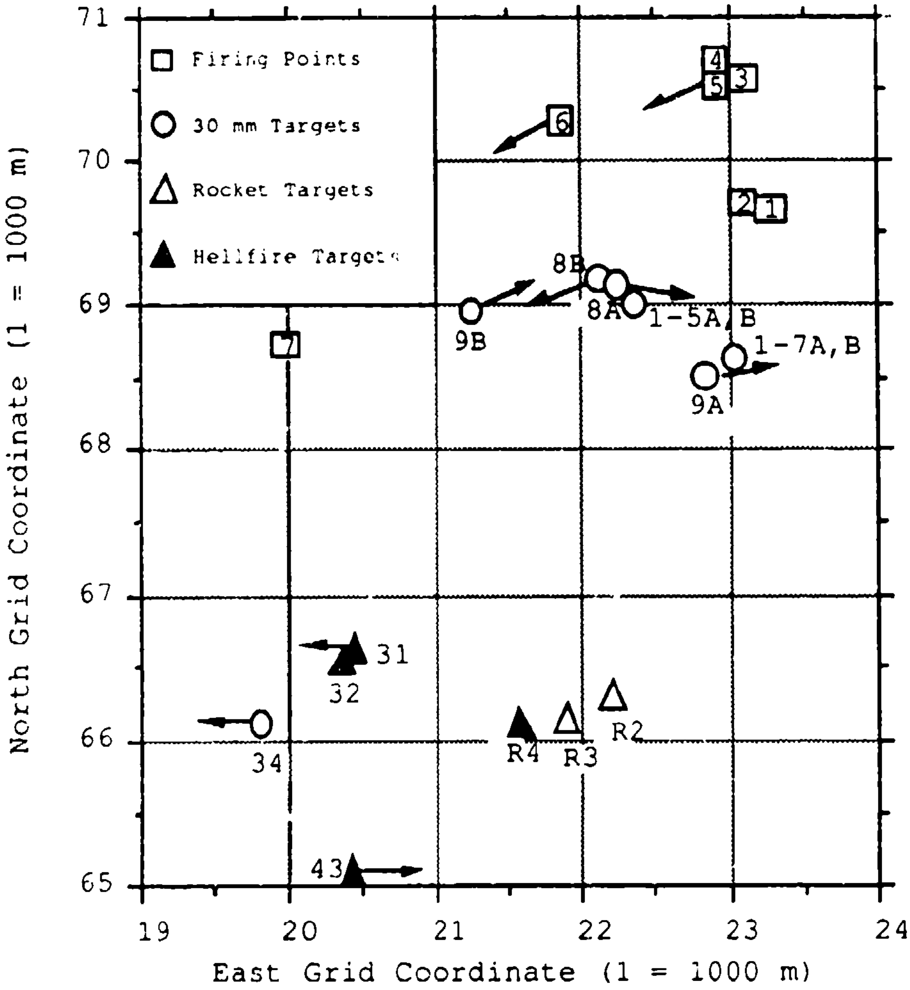


Figure 6. Configuration of the firing points and targets at the Dalton-Henson Multipurpose Range Complex, Fort Hood, Texas.

Table 1, but 30 mm target distances are greater than in Table 1 because engagements shorter than 2000 m are difficult to create in the simulator. Additionally, the missile engagement distances in the simulator are longer than in Table 1 because engagements longer than 5000 m are difficult to create on the live-fire range.

The tactical scenario was conducted with a temperature of 15°, a visibility of 7600 m , a ceiling of 3000 ft, a wind of 5 kts at 300°, and a barometric pressure of 29.92 in. The CMS threat lethality was set to 5 with hostility interrupt on. The visual mode (VM) and scene illumination (SI) were changed to simulate day and night (Day: VM = 2, SI = 5; Night: VM = 1, SI = 11).

Personnel

The three types of personnel participating in this research (AH-64A aviators, CMS I/Os, and range scoring personnel) are described in the following sections.

AH-64A aviators. Initially, 30 qualified and current AH-64A crews (60 aviators) were selected to serve as subjects for the study. All crews from three squadrons of an operational cavalry brigade who were scheduled to remain in the unit for at least 6 months were selected to participate. Because Army policy restricts females from gunship operations, all aviators were male. The experimenter, with the assistance of brigade and squadron SIPs, formed two matched groups on the basis of qualitative estimates of aviator experience and skill. Fifteen crews were assigned to the experimental (simulator) group and fifteen crews were assigned to the control (no simulator) group.

Before U.S. troop deployment to the Persian Gulf, crew attrition was minimal (3 crews) and unrelated to crew performance (i.e., permanent change of station, medical grounding). An additional 9 crews were lost to operational units of the Central Command before the final performance tests. Fortunately, crew loss was equal between the groups. At the conclusion of the research, 18 crews participated in the final live-fire exercises, 9 in each group. After completing the day run, however, one crew in the control group was unable to complete the night run or CMS test because of an off-duty injury to one crewmember.

During the initial live-fire exercises, demographic and flight experience information was obtained from the participating aviators using the AH-64 CMS Gunnery Research Program Demographic Survey (see Appendix A). The demographic data obtained from the survey indicate a range of experience that is typical of AH-64 operational units. Namely, the units consist of aviators with two distinctly different backgrounds: those with previous career experience in other helicopters (predominantly the AH-1) and those who proceeded from initial entry rotary wing training to the AH-64 Aviator Qualification Course (AQC). Analysis of the demographic data for the aviators who completed the research indicate that the training groups were similar when the research began (see Table 2). The differences that were found between the

Table 2

Aviator Demographic Data at the Initial Live-Fire Exercise

		Pilot		Gunner	
Measure	Quantity	Control (*n* = 9)	Simulator (*n* = 9)	Control (*n* = 9)	Simulator (*n* = 9)
Age (years)	median	30	31	30	26
	range	(23-40)	(25-47)	(26-34)	(22-40)
Months of Active Duty	″	118	120	78	80
		(20-213)	(37-267)	(46-153)	(18-216)
Months Since AQC	″	27	30	18	6
		(3-34)	(9-60)	(5-30)	(2-22)
AH-64A Flight Hours	″	541	434	288	218
		(229-622)	(148-788)	(149-638)	(149-518)
Total Flight Hours	″	1230	1168	761	416
		(386-4360)	(313-5940)	(382-2011)	(314-1727)
Readiness Level	mean	1.2	1.2	1.1	1.8
	SD	(0.44)	(0.67)	(0.33)	(0.83)
Range Experience	″	3.1	2.8	1.4	0.7
		(1.90)	(1.79)	(1.24)	(1.41)

Note. AQC = AH-64A Aviator Qualification Course. Readiness Level (RL) progresses from RL3 (new assignment to unit) to RL1.

simulator and control groups are small, especially when compared to the differences between the crew seat position. However, the AH-64A flight hours, readiness levels, and previous Dalton-Henson range experience indicate that the simulator group was somewhat less experienced than the control group.

CMS instructors. The gunnery instruction and console operation for crew testing and training in the CMS was conducted by seven civilian Flight Simulator Facility AH-64A CMS Instructor Pilots (IPs). All seven were retired Army IPs and were highly experienced in the CMS operation and instruction. The I/Os were briefed on the purpose, design, and procedures of the research project and participated in designing the tactical CMS gunnery scenario.

Scoring personnel. Target effect measures of gunnery performance were obtained during the live-fire exercises by nine individuals: four AWSS operators, three squadron SIPs, and two researchers. Two civilian contract personnel operated the AWSS during the day exercises and another two during the night exercises; on each shift, one scorer operated the BSS and one operated the DSS. The missile target effect performance was evaluated by each squadron's SIP. The researchers monitored the range activities, collected the performance information from the BSS and DSS operators, and entered the data into the project computer, one during the day exercises and the other during the night exercises. The researchers obtained engagement time measures for the live-fire exercises and the CMS tests from the VRS videotapes. They also obtained target effect measures for the CMS tests from computer-generated printouts.

Detailed Procedures

Live-fire exercises. The initial and final live-fire exercises were conducted at the Dalton-Henson Multipurpose Range Complex at Fort Hood, Texas. The initial live-fire exercises were conducted at two different times. Two squadrons from the participating unit completed the initial exercises over a 9-day period. The last squadron completed the initial exercises over a 5-day period approximately 2 months later. All squadrons completed the final live-fire exercises over a 15-day period 6 months after the first initial live-fire exercise. During both the initial and the final live-fire exercises, only one squadron occupied the range at a time. The experimental protocol for the live-fire exercises was similar for the initial and final exercises.

The gunnery exercises were controlled from the range operating tower. Each squadron provided one range safety officer and one communications (COM) officer. The range operations office provided one civilian to operate the automated range. All targets were raised and lowered under the computer control of the range operator in the tower.

Each squadron established a forward arming and refueling point (FARP) within one mile of firing point 1. For the entire period that the squadron occupied the range, unit personnel manned the bivouac for rearming, refueling, maintaining, and staging aircraft. Aircraft began and ended each run at the FARP. Each crew contacted the tower COM officer when they were ready to start a run. When the range was clear of preceding aircraft, the aircraft were cleared by the COM officer to move from the FARP to the first firing point.

Typically, a crew arrived on the range at firing point 1 and proceeded through firing point 7 in sequential order. If equipment malfunction or other problems occurred, the crews were instructed to return to the FARP to obtain aircraft maintenance or replacements. Subsequently, the crews returned to the range to complete all engagements. Performance of all the gunnery tasks in Table 1 and, consequently, progress through all seven firing points was referred to as a run. Each crew completed one run under day conditions and one run under night conditions. During the initial exercises, crews were allowed to complete multiple runs to pass unit standards for gunnery performance. Shortages of range time and ammunition during the final exercises limited each crew to a single day and a single night run.

All aircrews followed standard out-front boresight procedures before firing the aircraft laser or weapons. Upon arriving at each firing point, the COM officer acknowledged the aircraft's arrival at the firing position, cleared the crew to arm the weapon systems, instructed the crew to activate the VRS, and randomly selected one of the target engagements defined for that firing point. For each engagement, the COM officer performed the following activities:

- requested that the range operator raise the target;
- requested that the aircraft establish the minimum safe altitude of 50 ft above ground level (AGL); and
- delivered a standard target handover including bearing, description, mode (stationary or moving), and weapon.

After receiving the target handover, the crew performed the following activities:

- established an altitude of 50 ft AGL,
- acknowledged the target handover,
- positioned switches for the engagement,
- unmasked the aircraft,
- acquired the target,
- delivered the ordinance,
- masked the aircraft, and
- called "weapons clear" to the COM officer.

When the crew called weapons clear, the COM officer instructed the crew to deactivate the VRS and to place the weapon systems in the safe mode; he then cleared the crew to proceed to the next firing position.

During the initial and final live-fire exercises, each crew was allowed to choose the weapon mode used to engage each target. However, the crews consistently used the same mode, which probably represented the consensus on the optimal weapon system mode for each engagement. The 30 mm engagements were conducted by the CPG using the TADS and LRF/D. Rocket engagements were conducted in the cooperative mode: The CPG tracked the target with the LRF/D and TADS and the PLT maneuvered the aircraft to align the rocket symbology and fire the weapon. Missile engagements were conducted using the aircraft's simulated Hellfire training missiles by the CPG using the TADS and LRF/D in a normal lock-on-before-launch mode with autonomous target designation.

AH-64A CMS test procedures. The CMS scenario was used to test the gunnery performance of all crews after the initial live-fire exercises and again after the final live-fire exercises. The CMS was used in the integrated mode both for testing and training gunnery performance. The CMS gunnery performance test was conducted for both day and night conditions during a 1.5-hour simulator period.

Each crew arrived at the simulator facility 30 to 40 minutes before the scheduled simulator session. When the crew arrived, the I/O gave them a copy of the situation sheet, a tactical map, a contour chart, and a communications frequency list. The crews were then allowed to plan the mission before the simulator session began; they could obtain assistance from the I/O, if necessary.

Each crew began the scenario in a holding area and flew to the first firing position under the direction of the scout, who was played by the I/O. From the first firing position, the crew fired missiles, rockets, and 30 mm rounds at different targets. The scout then directed the crew to move to another firing position, where the crew engaged other targets using the missiles and rockets. Subsequently, the scout directed the crew to move to a grid point. When the crew arrived at the grid point, the scout directed the crew to proceed cautiously in the direction of another grid point to assist in locating a downed friendly aircraft. As the aircraft traveled through the lowland route, the scout called for the crew to suppress a target using the 30 mm gun. When the aircraft arrived at the second grid point, the scout instructed the crew to turn around and make another reconnaissance pass over the lowland route and to engage the target again using the 30 mm gun. After completing the engagement, the scout directed the crew to proceed to another highland battle position, where a final missile target was engaged.

After the crews completed the scenario under day conditions, they repeated it under night conditions. When the test session was completed, the crew reviewed their performance with the I/O and returned the test materials. The time required for each test session was approximately 2.5 hours.

The CMS VRS was used during the test to record all engagements. During each test, the I/O directed the crews through the scenario by acting as the scout. He did not provide any instruction or performance feedback to the crews during the CMS test. As each target was engaged, the summary of the ownship gunnery performance, generated by the CMS, was printed by the researcher.

Experimental group training procedures. After the initial live-fire exercise and CMS test, the squadron and brigade SIPs were instructed to continue the normal unit training of the simulator group aviators, with the following three exceptions. First, the simulator group aviators were restricted from participating in any live-fire gunnery practice in the aircraft. Second, they were instructed to complete the Postflight Debriefing form after each flight in the aircraft, CMS, or CWEPT and to submit the completed forms periodically to the on-site researcher. Third, they were required to attend five gunnery training sessions in the CMS before the final live-fire exercise.

The experimental group's gunnery training was conducted exclusively in the CMS. The procedures for training crews in the CMS were similar to those used in the CMS gunnery tests with two exceptions. First, the VRS was not used during CMS training. Second, the I/O aided and instructed the crewmembers as necessary during the mission.

Control group training procedures. After the initial live-fire exercise and CMS test, the squadron and brigade SIPs were instructed to continue the normal unit training of the control group aviators, with the following two exceptions. First, the control group aviators were restricted from gunnery practice in the CMS, but they were allowed to use the CMS for instrument and emergencies procedures training. Second, they were instructed to complete the Postflight Debriefing form after each flight in the aircraft, CMS, or CWEPT and to submit the completed forms periodically to the on-site researcher. Except for the initial and final gunnery tests, the researchers had no direct contact with the control group aviators.

Monitoring Procedures

The use of the CMS was the only training activity under the experimental control of this research. However, there are several other training activities that could significantly affect crew gunnery performance. Differential use of other forms of gunnery training by the two groups could confound the results of the research. Therefore, participation in JAAT training exercises and the use of the aircraft, the CWEPT, and the TADS Selected Task Trainer (TSTT) were monitored over the course of the research to aid in the interpretation of the results. Squadron operations officers provided information about major gunnery training activities (e.g., JAATs). The Army aviator flight records (Form 759) were reviewed after the initial and final live-fire exercises to measure the amount of AH-64A flight time. Finally, the on-site researcher obtained the number of hours that the participating aviators used the CWEPT and TSTT from a computer data base maintained by personnel at the simulator facility.

Measures of Effectiveness

Several measures of effectiveness (MOEs) were obtained during the live-fire and CMS gunnery performance tests. When more than one run was completed by a crew during the initial live-fire exercise, the performance on the last run completed was used. With the exception of engagement time, the MOEs differed from one weapon system to another and from the live-fire exercises to the CMS tests. Each of the measures and their source are described in the following sections.

Engagement time. Engagement time was defined as the time between when the crew acknowledged the target handover and when they called weapons clear. The VRS was used during the live-fire exercises and the CMS tests to record TADS displays during each engagement. All engagement time measures were obtained using hand-held stop watches and the VRS videotapes after the exercises. The live-fire range and CMS protocol were designed to utilize the 1-hour videotapes efficiently and to provide objective start and finish events to aid in measuring engagement time.

30 mm target effect. For the live-fire exercises, 30 mm target effect was defined as the number of rounds that passed through the BSS Doppler fan and landed in the target effect area (hits) divided by the total number of rounds fired from the aircraft (shots). The number of hits was provided by the BSS operator and the shots were obtained from the rounds

counter on the VRS videotape of the engagements. Thus, the hits/shots ratio is the percentage of rounds in the target effect area or box.

Rocket target effect. For the live-fire exercises, rocket target effect was defined for each 4-rocket engagement as the mean distance from the target (miss distance). The down-range miss distance and the cross-range miss distance for each rocket impact was provided by the DSS operator. For each rocket impact sensed by the DSS, the cross- and down-range miss distances were used to compute the absolute miss distance using the Pythagorean theorem. Because the DSS demonstrated good sensitivity for rocket impacts out to 350 m, all rocket impacts that were not detected by the DSS were assigned 500 m miss distances by the researcher.

Hellfire target effect. During the live-fire exercises, the VRS videotapes were viewed immediately after each run by the squadron SIP and evaluated using the brigade standard for missile target kills. The information taken from the tapes was used to evaluate proper mode selection, switch settings, target acquisition, missile launch, and guidance. The squadron SIPs recorded whether the target was killed on brigade evaluation sheets.

CMS target effect. The ownship performance data sheets generated by the CMS after each engagement were the source of the target effect measures for 30 mm, rocket, and missile performance in the simulator. For each weapon trigger pull, the CMS calculated the mean miss distance for the rounds fired. If any rounds from a trigger pull hit the target, the mean miss distance was always zero. The mean distances for each trigger pull were used to compute the mean miss distance for each engagement by creating a rounds-weighted sum of miss distance and then dividing by the total number of rounds fired. In addition to miss distance, target impacts (kills) were recorded for each engagement.

Results

The first major objective of this research was to determine the effectiveness of the CMS for the sustainment of crew gunnery skills. CMS effectiveness was determined by analyzing the live-fire gunnery exercises, the CMS gunnery test, and the other training activities.

Live-Fire Gunnery Performance Test

The effectiveness of the CMS was directly tested by comparing the performance of the training groups during the pretest and the posttest live-fire exercises. Because the simulator could have differential effectiveness across weapon systems, the training effectiveness of the CMS was analyzed separately for each weapon. Further, the training effectiveness of the CMS was analyzed separately for measures of target effect and engagement time to determine if the simulator had a differential effect on the two aspects of gunnery performance.

Two-factor repeated measures analysis of variance (ANOVA) tests were conducted on each dependent measure. Training (simulator vs. control) was analyzed as a between-group variable. Trial (initial vs. final exercise) was analyzed as a within-group, or repeated measures, variable. In this ANOVA design, transfer of training is indicated by a significant interaction between training and trial. Positive transfer is indicated when the simulator group performs better than the control group during the final exercise. Ideally, the gunnery performance of both groups would be equivalent at the initial live-fire exercise (matched groups) and differ at the final live-fire exercise. A trial main effect would indicate significant changes in the performance across trials unrelated to training group. A training main effect indicates a lack of equivalence between the groups across trials.

The results from the six live-fire analyses (three weapon systems by two measures) are presented in the following paragraphs. All of these analyses were initially conducted separately for day and night. In no case, however, did the trends found for day or night differ from the combined trends. To simplify the presentation of the results, only the analyses of the data combined across day and night are reported.

Finally, for each of the analyses presented below, the gunnery performance measures are graphed. Each graph displays the mean and one standard error of the mean (plus and minus) for each training group during the initial and final live-fire exercises. The standard error of the mean quantifies the variability in the data and, when graphed, provides a visual indication of the differences in the individual scores and the significance of the differences between the means. Means with standard error bars that overlap are generally not significantly different from one another; means with nonoverlapping error bars usually differ significantly.

Engagement time. The times for the 30 mm gun and Hellfire missile engagements were similar and averaged 69 and 66 seconds, respectively (see Figures 7 and 8). In contrast, the rocket engagement times were substantially longer, averaging 159 seconds per engagement (see Figure 9). This difference was the result of the requirement that each of the four rockets in each engagement must be fired individually. The standard errors of the mean are shown as vertical bars in all the figures.

There were no significant interaction effects for any of the engagement times, but there were differences in the trends shown for the three weapon systems. The engagement times for the 30 mm gun (see Figure 7) indicate that the simulator group improvement was slower than the control group improvement over the course of the experiment (i.e., negative transfer). However, both the missile and rocket data demonstrate a trend toward positive CMS transfer (see Figures 8 and 9).

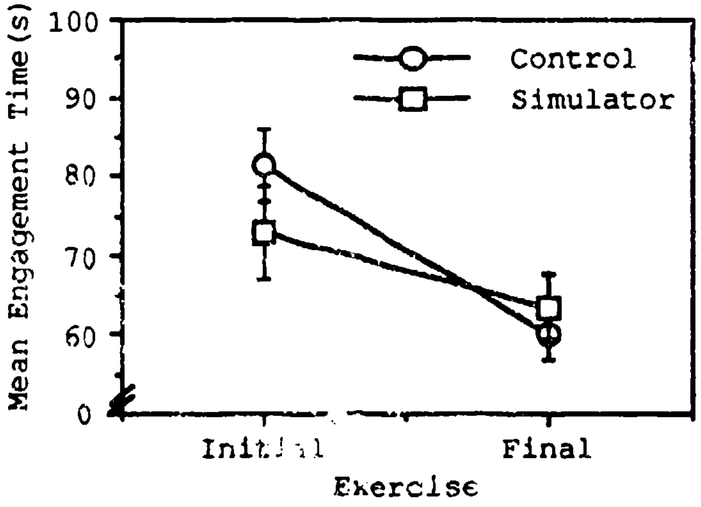


Figure 7. The mean 30 mm engagement time ± 1 standard error during the initial and final live-fire exercises as a function of training group.

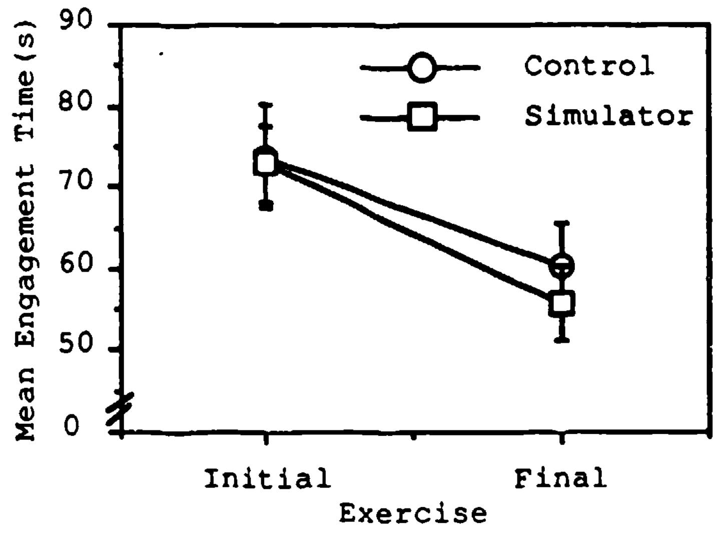


Figure 8. The mean Hellfire engagement time ± 1 standard error during the initial and final live-fire exercises as a function of training group.

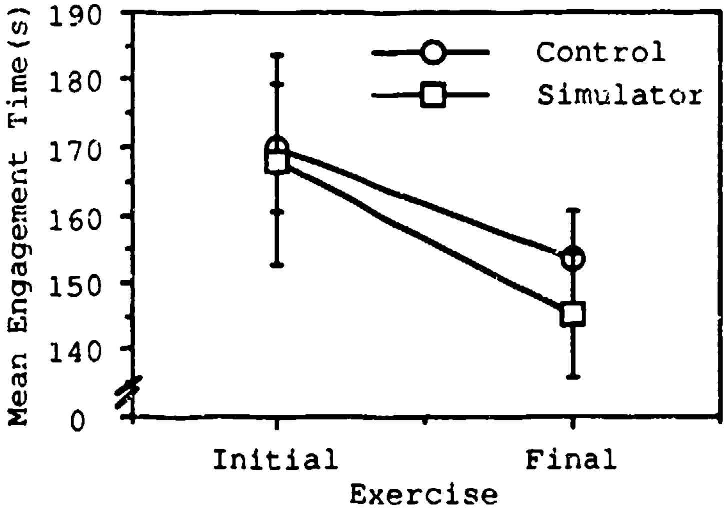


Figure 9. The mean rocket engagement time ± 1 standard error during the initial and final live-fire exercises as a function of training group.

There were significant improvements in both groups between the initial and final live-fire exercises (i.e., trial main effect) for the 30 mm gun (F (1, 15) = 13.53, p <.05) and the Hellfire missile (F (1, 15) = 11.80, p <.05). This 15-second improvement may be due to the practice received during the initial live-fire exercise or some other non-CMS training that occurred between the initial and final live-fire exercises. A similar improvement of 20 seconds in the rocket data was not significant, however, probably because of the large amount of variance within the groups (see the standard error bars in Figure 9).

Overall, the engagement time data show no significant effect on CMS training. The engagement data, however, do demonstrate a consistent (≈17%) improvement over the course of the experiment.

Target effect. Averaged across firing points, the 30 mm gun performance for all groups was approximately 50% (see Figure 10). Though there were strong range-to-target effects in the 30 mm target effect data (see Hamilton, 1991), there were no significant CMS training effects. The performance of the simulator and control groups was almost identical during the initial exercise, but the control group performed slightly better than the experimental group during the final live-fire exercise.

The mean miss distance for rockets varied from approximately 350 m to 250 m during the experiment (see Figure 11). The mean miss distance for the control group was significantly better than the simulator group during both exercises (F (1, 15) =15.26, p <.05). However, the ANOVA did not indicate a CMS training effect (i.e., trial by training interaction). There was also a significant improvement in mean miss distance of approximately 50 m from the initial to the final live-fire exercises (F (1, 15) = 6.66, p <.05). This effect can probably be attributed to improvements in rocket pod alignment techniques implemented between the initial and final exercises (see Hamilton, 1991).

Finally, the Hellfire performance was quite high: The crews always scored at least 9 of the 16 possible missile kills. Missile kill performance was nearly identical at the initial exercise, but the control group performance was slightly better than the simulator group performance during the final live-fire exercise (see Figure 12). However, there were no significant differences in the Hellfire missile performance.

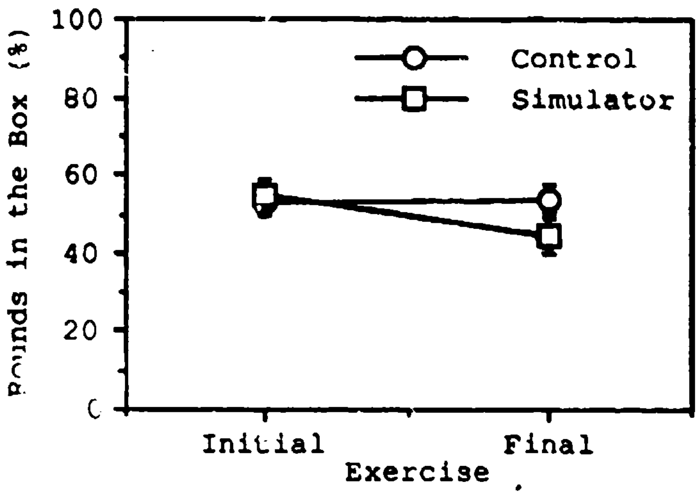


Figure 10. The percentage of hits ± 1 standard error for the 30 mm gun during the initial and final live-fire exercises as a function of training group.

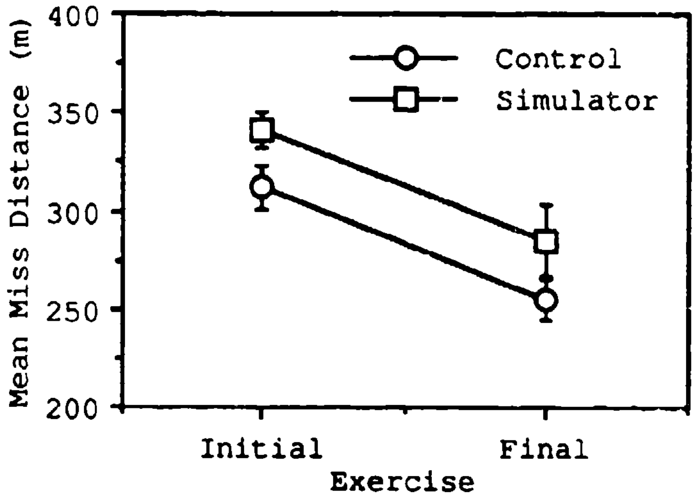


Figure 11. The mean miss distance ± 1 standard error for the rockets during the initial and final live-fire exercises as a function of training group.

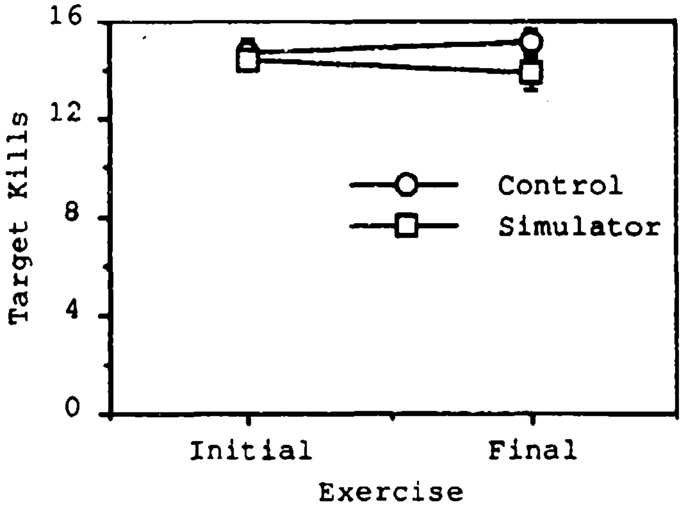


Figure 12. The percentage of judged target kills ± 1 standard error for the Hellfire missiles during the initial and final live-fire exercises as a function of training group.

Overall, the live-fire gunnery range results show little evidence of CMS training effectiveness in the simulator group or of skill decay in the control group. There were several examples of initial-to-final performance improvements, but the trends were not related to CMS training.

CMS Gunnery Performance Test

The analysis of the effect of simulator training on gunnery performance in the CMS was conducted using the same ANOVA design used for the live-fire data. Examination of the differential performance in the CMS is a test of training effectiveness as opposed to simulator effectiveness. The control group was expected to show some skill decay; the simulator group was expected to show some skill enhancement.

As in the live-fire analyses, engagement time and target effect measures (mean miss distance and number of target kills) were analyzed separately for each weapon system. The target effect measures generated by the CMS are classified and cannot be reported in detail. However, the overall trends found during these analyses are sufficient to evaluate the effect of the simulator training on simulator performance.

Overall, there were no significant effects at the $p < .05$ level for nine ANOVAs (3 weapons by 3 measures), but the gunnery performance trends in the simulator were more consistent than those found in the live-fire exercises. Gunnery performance of the simulator group improved for each of the measures across all weapons systems and, with one exception, improved by a greater amount than the control group. Moreover, some indication of skill decay was found in the control group for both rocket target effect measures.

Other Training Activities

Data were collected and analyzed for differential use by the two training groups for four types of training: JAAT, aircraft, CWEPT, and TSTT. The aviators were instructed to complete a Postflight Debriefing form after every training activity, but they were very inconsistent in complying with this requirement. As a result, the information about non-CMS training activities is drawn only from more reliable sources.

First, the unit involved in the research participated in a JAAT training exercise at Fort Hood, Texas, 2 months before the final live-fire exercises. Four of the control group crews participated in the training, but no additional information is available about the type or amount of training they received. Second, the mean flight hours per crewmember during the experimental period were larger in the control group ($M_{NoCMS} = 127$, $SE = 10.7$) than the simulator group ($M_{CMS} = 101$, $SE = 8.0$), but the differences were not statistically significant. The greater number of aircraft flight hours in the control group may be partially attributed to the JAAT exercise.

Third, CWEPT records indicated the participating crews did not use the device very often and there was no statistically significant difference between the average number of hours each group used the device ($M_{NoCMS} = 1.31$, $SE = .368$; $M_{CMS} = 1.25$, $SE = .829$). Finally, a TSTT training device was available to participating crews during the initial stages of the research, but it was removed approximately 3 months before the final live-fire exercises. Discussions with personnel managing the device again indicated little or no use of the device by the participating crews.

Discussion

The first major objective of this research was to determine the effectiveness of the CMS for the sustainment of crew gunnery skills. The effectiveness of the CMS was directly tested by measuring the differential performance of the training groups between the initial and final live-fire exercises. The results of these tests can be summarized in three statements.

First, after five gunnery training sessions in the CMS, the simulator crews did not have significantly better engagement time or target effect performance when compared to the control group. Second, the results of the initial and final CMS gunnery performance tests showed consistent but nonsignificant performance improvements in the simulator group. Third, there were no significant differences in the other training practices of the simulator and control group aviators. Though these findings appear to indicate that the CMS is ineffective in sustaining gunnery skills, a number of factors should be considered before drawing final conclusions from the research results.

Skill Decay

The best demonstration of training effectiveness for skill sustainment is for the control group to show skill decay while the simulator group maintains their skill level. The results of this research show no sign of skill decay by the control group. Indeed, the performance of the control group improved in many instances over the course of the research. As anticipated in the design considerations, a measurable loss of skill in the control group would be required to demonstrate skill sustainment, and thus, CMS training effectiveness.

The primary reason that skill decay was not observed in the control group is probably the short time span of the research. The minimum length of time for aviator skill decay has been shown to be at least 6 months (Ruffner & Bickley, 1983; Ruffner, Wick, & Bickley, 1984). However, appreciable skill loss probably occurs sometime between 6 months and a year for aviators not engaged in any form of gunnery training. Skill decay in the control group may also have been minimized by factors such as participation of the control group in the JAAT training exercises, aircraft dry-fire exercises, the initial CMS test, or simple mental rehearsal.

Skill Enhancement

An alternative method of demonstrating training effectiveness for skill sustainment is for the simulator group to show skill enhancement while the control group maintains their skill level. The research results demonstrate that CMS training over a 6-month period was not sufficient to make the simulator group's skill measurably better than the control group's. This finding probably indicates that the participating aviators were highly proficient when the research began. Ironically, the high level of gunnery proficiency may be the result of a successful unit training program that included the fielding, staffing, and effective use of CMS facilities in the unit.

Another explanation of the lack of skill enhancement may be that the measures of effectiveness were not sufficiently sensitive to detect increases in the aviator's gunnery proficiency. Sensitive measures are difficult to identify because of the amount of variability introduced by random variables such as different aircraft, aircraft maintenance, and weather. Nonetheless, the results from other analyses conducted to evaluate the AH-64A gunnery standards indicate that the performance measures were sensitive to several factors other than aviator training, including changes in range to target and differences in aircraft weapons maintenance procedures (see Hamilton, 1991). As a result, the lack of skill enhancement is more likely attributable to high initial skill levels than to insensitivity in the measures of effectiveness.

Conclusions

The results of this research support three conclusions related to the first objective of this research, to determine the effectiveness of the CMS for sustaining crew gunnery skills. First, this experiment found no significant positive or negative transfer of training from the CMS to the live-fire gunnery range. Thus, the training effectiveness of the CMS to sustain crew gunnery skills remains equivocal.

Second, the gunnery proficiency of operational AH-64A aviators is at or near an asymptotic level of performance. For this reason, the CMS did not substantially improve aviator performance during the 6-month period of this experiment. However, there is no evidence that monthly CMS training produces any negative transfer to the aircraft.

Third, the restriction from CMS training was insufficient to bring about skill decay in the control group over the 6-month period of this research. Although previous research indicated that skill decay could occur within 6 months, there was no evidence of skill loss in the control group during the final live-fire exercises. Thus, current levels of aircraft and other forms of gunnery training are sufficient to maintain crew gunnery skills in proficient aviators for up to 6 months without the aid of CMS training.

The second objective of this research was to provide data to establish an optimum combination of aircraft and flight simulator training for the sustainment of crew gunnery skills. Because the research did not establish the benefits of short term CMS training or the critical period for gunnery skill decay, the obtained data are insufficient to determine the optimal use of the CMS for gunnery training of operational aviators.

Recommendations

Until additional empirical data can be obtained, the CMS should remain an integral part of operational gunnery training. The results indicate that if aircraft hours and other gunnery training (e.g., JAATs) are funded at the levels observed in this research, the critical period for gunnery skill decay in proficient aviators is 6 months or longer. Thus, CMS gunnery training may be required only biannually. Nevertheless, a more conservative quarterly CMS gunnery training may be advisable, especially when the participating aviators may have benefited from the initial live-fire and CMS gunnery tests. However, if the support for aircraft hours and other gunnery training exercises is reduced, gunnery skills may decay in less than 6 months unless CMS training is increased. In fact, there is no statistically significant evidence of negative transfer when CMS gunnery training is conducted on a monthly basis.

Research Limitations

As with any research, the application of the results of this experiment is limited by the conditions under which they were obtained. Although there are others, the four major limitations brought about by the selection of sustainment training, gunnery training, crew training, and the measures of effectiveness are discussed in the following paragraphs.

First, the research was designed to measure the training effectiveness of the CMS for sustaining skills. As such, it produced information pertinent to unit sustainment training but not to the acquisition of skills. Thus, the five CMS training sessions that did not produce positive transfer to the aircraft for proficient aviators might significantly improve the performance of unskilled aviators. Second, the research focused on gunnery skills; different results may be obtained for other skills such as instrument flight and emergency procedures.

Third, the research addressed only the crew level of gunnery training. The Army has structured the gunnery training of its aviators in a logical progression from the acquisition of individual skills, through crew skills and coordination, to team skills and coordination. The effectiveness of the CMS may be different for the other levels of gunnery training.

Fourth, the MOEs used in this research further limit the generalizability of the results. Although speed and accuracy are classic measures of gunnery skill, many other skills are critical to the success of helicopter gunnery missions. One example is the identification, selection, and use of terrain to mask the helicopter from enemy threat. Because the criterion for selecting firing points on the gunnery range and for selecting battle positions during a gunnery mission differ significantly, appropriate terrain masking techniques were not emphasized during this research project. However, terrain masking is a tactical gunnery skill that the CMS may be effective in training.

Future Research

The costs of AH-64A gunnery training resources (e.g., flight and range time, ammunition) have increased the Army's dependence on flight simulators for training that was previously accomplished in the aircraft. Most Army aviators are required to accomplish a portion of their annual flight requirements in a flight simulator. Furthermore, the trend toward substituting simulator training for aircraft training is likely to continue as resources become more expensive and simulator technology becomes more advanced.

The Army has not based the deployment or utilization of flight simulators on empirical training effectiveness data that relate to the acquisition or sustainment of gunnery skills. In fact, individual unit commanders are responsible for determining the mix of aircraft, simulators, and other training devices that make up their training program. Even when two or more units share the same simulator site, there are differences in the ways that units use the flight simulators. Decisions about the trade-off between aircraft and simulator time should be based on empirical demonstrations of the simulator's effectiveness for training specific tasks. Commanders could use this information to develop training programs that achieve their training goals and maximize the utilization and effectiveness of the training resources available.

This research represents an initial step toward empirically determining the effectiveness of the CMS for satisfying the gunnery training requirements of operational aviation units. The results of this research add significantly to the knowledge base about the time course of AH-64A gunnery skill decay and sustainment in operational units, but many questions remain to be answered.

Thus, further investigations of gunnery skill decay in proficient aviators should be conducted over a longer period of time, such as 12 to 18 months. The research should be designed to establish the relative effectiveness of each of the alternative training devices currently available to operational units for sustaining gunnery skills. If sufficient control can be maintained during the proposed research, the information necessary to design an efficient training strategy could be determined.

Because good gunnery and tactical skills affect crew survivability, research that requires the significant loss of those skills may be unethical. In the design of future research, control groups should be identified whose lives would not be endangered by a discontinuation of gunnery training (e.g., aviators retiring from active duty, aviators assigned to nonflying duties).

Finally, future research should be given adequate fiscal, personnel, and operational support. The utility of the current research was severely limited by crew attrition and scheduling problems that must be resolved before satisfactory data can be obtained to address questions about sustaining AH-64 gunnery skills.

References

Bridgers, J. D., Bickley, W. R., & Maxwell, W. G. (1980). Operational test II of the AH-1Q flight and weapons simulator (OTN-139). Fort Rucker, AL: U.S. Army Aviation Board.

Cross K. D., & Gainer, C. A. (1987). An enumeration of research to determine the optimal design and use of Army flight simulators (ARI Technical Report 763). Alexandria, VA: Army Research Institute for the Behavioral and Social Sciences. (AD A194 242)

Department of the Army. (1984, June). Operator's manual for the AH-64 helicopter (Training Manual 55-1520-238-10). Washington, DC: Headquarters, Department of the Army.

Department of the Army. (1986). Helicopter gunnery (Field Manual 1-140). Washington, DC: Headquarters, Department of the Army.

Department of the Army. (1988, March). Operator's manual for the AH-64A (Apache) Combat Mission Simulator (Training Manual 55-6930-214-10). Washington, DC: Headquarters, Department of the Army.

Holman, G. L. (1979). Training effectiveness of the CH-47 **flight simulator (ARI Research Report 1209). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A072 317)**

Hamilton, D. H. (1991). AH-64A gunnery performance: Implications for gunnery standards (Interim Report AS690-342-91A). Fort Rucker, AL: Anacapa Sciences, Inc.

Hopkins, F. C. (1979). Cost and training effectiveness of the AH-1 flight simulator (ACN23881). Fort Rucker, AL: Directorate of Training Developments, U.S. Army Aviation Center.

Jacobs, J. W., Prince, C., Hays, R. T., & Salas, E. (1990). A meta-analysis of the flight simulator training research (Technical Report 89-006). Orlando, FL: Naval Training Systems Center Human Factors Division.

Kaempf, G. L., & Blackwell, N. J. (1990). Transfer-of-training study of emergency touchdown maneuvers in the AH-1 Flight and Weapons Simulator (ARI Research Report 1561). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A226 360)

Kaempf, G. L., Cross, K. D., & Blackwell, N. J. (1989). Backward transfer and skill acquisition in the AH-1 Flight and Weapons Simulator (Research Report 1537). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A213 432)

Luckey, J. R., Bickley, W. R., Maxwell, W. G., & Cirone, E. A. (1982). Operational test II Phase I of the UH-60A Black Hawk operational flight simulator (OTN 505). Fort Rucker, AL: U.S. Army Aviation Board.

McAnulty, D. M., & Kaempf, G. L. (1991). Effectiveness of the AH-1 Flight and Weapons Simulator (FWS) for sustaining aerial gunnery skills (ASI690-343-91B). Fort Rucker, AL: Anacapa Sciences, Inc.

Orlansky, J., & String, J. (1977). Cost-effectiveness of flight simulators for military training. Arlington, VA: Institute for Defense Analyses, Science and Technology Division.

Roscoe, S. N. (1971). Incremental Transfer Effectiveness. Human Factors, 13(6), 561-567.

Roscoe, S. N., & Williges, B. H. (1980). Measurement of transfer of training. In S. N. Roscoe (Ed.), Aviation Psychology (pp. 182-203). Ames, IO: The Iowa State University Press.

Ruffner, J. W., & Bickley, W. R. (1983). Validation of aircrew training manual practice interaction requirements (Technical Paper 696). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences. (AD A173 441)

Ruffner, J. W., Wick, D. T., & Bickley, W. R. (1984). Retention of helicopter flight skills: Is there a "critical period" for proficiency loss? In, Proceedings of the Human Factors Society, 28th Annual Meeting (pp. 370-374). Santa Monica, CA: Human Factors Society.

U.S. Army Audit Agency. (1982). Report of audit: Synthetic flight training system (Audit Report SO 82-6). Washington, DC: U.S. Army Audit Agency.

U.S. Army Audit Agency. (1985). Report of audit: Follow-up audit of synthetic flight training system (Audit Report SO 85-18). Washington, DC: U.S. Army Audit Agency.

U.S. Army Aviation Center. (1988, June). Helicopter gunnery - coordinating draft (Training Circular 1140). Fort Rucker, AL: U.S. Army Aviation Center.

U.S. Army Aviation Center. (1990, April). Helicopter gunnery - approved draft (Training Circular 1-140). Fort Rucker, AL: U.S. Army Aviation Center.

U.S. Army Research Institute Aviation Research and Development Activity. (1986). Flight simulator skill sustainment training research plan. Fort Rucker, AL: U.S. Army Research Institute Aviation Research and Development Activity.

Valverde, H. H. (1973). A review of flight simulation transfer of training studies. Human Factors, 15(6), 510-523.

APPENDIX A

AH-64 CMS GUNNERY RESEARCH PROGRAM DEMOGRAPHIC SURVEY

6th CAVALRY BRIGADE--AIR COMBAT (CBAC)

PART A INSTRUCTIONS: Part A consists of questions that provide information about your personal background and experience. Answer each item that applies to you by checking in the appropriate bracket [√] or by printing the required information in the space provided. When answering items about flight hours, you may refer to records, if available, or you may estimate the flight hours as closely as possible. Your responses will be used for research purposes only.

1. Name: ____
Last First Middle

2. Social Security Number: ___ ___ ___ - ___ ___ - ___ ___ ___ ___

3. Today's Date: ________________________________
(Month) (Day) (Year)

4. What is your age ?
________ Years

5. What is your current rank?
[ ] WO1
[ ] CW2
[ ] CW3
[ ] CW4
[ ] 2LT
[ ] 1LT
[ ] CPT
[ ] MAJ
[ ] LTC
[ ] COL

6. To which unit are you assigned?
Unit: ________________ Squadron ________________ Troop

7. How long have you been assigned to your present troop?
________ years and ________ months

8. Do you anticipate reassignment prior to October 1990?

[ ] Yes

[ ] No

If yes, give expected date and location of reassignment________________

____

9. Currently, what is your primary duty position in the unit?

____

10. What additional duties do you perform in your unit?

____

____

____

11. How long have you been on active duty military service?

________ years and ________ months of active service

12. How long has it been since you graduated from initial Army flight training?

________ years and ________ months

13. How long has it been since you graduated from the AH-64 AQC?

________ years and ________ months

14. Were you an IERW turnaround student in the AH-64 AQC?

[ ] Yes

[ ] No

If no, what was your primary aircraft before entering the AH-64 AQC?

____

15. Indicate the total number of flight hours you have logged in each of the following aircraft. Also, check [√] the highest duty category you have held in each aircraft.

a. Military Rotary Wing

		PI	PC	UT	IP	SI	IE
AH-64: ________	hours	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
AH-1: ________	hours	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
OH-58: ________	hours	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
UH-1: ________	hours	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Other: ________	hours	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

(Specify other aircraft)________________________________

b. Military Fixed Wing

UH-21: ________	hours	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
C-12: ________	hours	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
OV-1: ________	hours	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]
Other: ________	hours	[ ]	[ ]	[ ]	[ ]	[ ]	[ ]

(Specify other aircraft)________________________________

16. How many flight hours have you logged in each seat of the AH-64?

Front Seat:__________ hours

Back Seat:__________ hours

17. How many flight hours have you logged in each seat of the AH-64 CMS?

Front Seat:__________ hours

Back Seat:__________ hours

18. How many flight hours have you logged in each seat of the AH-64 CWEPT?

Front Seat:__________ hours

Back Seat:__________ hours

19. How many night flight hours have you logged in each seat of the AH-64?

Front Seat:__________ hours

Back Seat:__________ hours

20. If you were an AH-1 pilot previously, how long has it been since you completed the AH-1 Crew Gunnery Tables?

________ years and ________ months

21. After arriving at your present unit, what was your original crew station designation?

[ ] AH-64 front seat

[ ] AH-64 back seat

[ ] Other (explain) ______________________________

22. How many training hours were required for you to attain RL2 and RL1 status in your originally designated seat? (Check here [ ] if you did not attain RL2 or RL1 in your originally designated seat.)

__________ flight hours to RL2 from RL3

__________ CMS hours to RL2 from RL3

__________ flight hours to RL1 from RL2

__________ CMS hours to RL1 from RL2

23. What is your current crew station designation?

[ ] AH-64 front seat

[ ] AH-64 back seat

[ ] Both seats (explain) ______________________________

24. What is your current Readiness Level?

RL Front	RL Back
[ ] RL1 in the front seat	[ ] RL1 in the back seat
[ ] RL2 in the front seat	[ ] RL2 in the back seat
[ ] RL3 in the front seat	[ ] RL3 in the back seat

25. Excluding IP evaluations, how many crewmembers have you flown with since entering the 6th CBAC?

__________ crewmembers

26. Have you been assigned to a fixed crewmate?

[ ] Yes

[ ] No

27. If you are a member of a fixed crew, how many hours has your crew trained together?

___________ flight hours

___________ CMS hours

28. How many of your flights, if any, have been delayed or rescheduled due to the unavailability of an appropriately trained (i.e., current in the required seat) crewmate?

___________ flights have been delayed or rescheduled

29. How many times have you participated in gunnery exercises at the Dalton/Henson range complex?

___________ times flying the AH-64A

___________ times flying other aircraft

6th CAVALRY BRIGADE--AIR COMBAT (CBAC)

PART B INSTRUCTIONS: Part B consists of questions that provide information about your experience with the AH-64 optical systems. The questions ask for both objective and subjective information. Answer each item that applies to you by checking in the appropriate bracket [√] or by printing your answer in the space provided. This information will be treated as highly confidential; individual responses will not be seen by anyone except the research staff.

30. In the back seat, how often do you optimize your PNVS FLIR?

[ ] Only during preflight checks
[ ] Rarely during flight
[ ] Occasionally during flight
[ ] Frequently during flight

31. In the front seat, how often do you optimize your TADS FLIR?

[ ] Only during preflight checks
[ ] Rarely during flight
[ ] Occasionally during flight
[ ] Frequently during flight

32. In the back seat, to what extent does the flight symbology interfere with your ability to see terrain features during NOE flight?

[ ] Not at all
[ ] Slightly
[ ] Moderately
[ ] A great deal

33. In the front seat, to what extent does the TADS weapons symbology interfere with your ability to see targets?

[ ] Not at all
[ ] Slightly
[ ] Moderately
[ ] A great deal

34. How difficult is it to read the numbers on the helmet-mounted display?

[ ] Not at all difficult

[ ] Slightly difficult

[ ] Moderately difficult

[ ] Very difficult

[ ] Extremely difficult

35. How much practice is required to handover targets proficiently between crewmembers using the flight and weapons symbologies?

[ ] Only initial practice

[ ] Occasional practice

[ ] Frequent practice

[ ] Constant practice

36. In your opinion, how likely are there to be misinterpretations of the different symbologies on the PNVS and TADS as a result of changing crew stations?

[ ] Not at all likely

[ ] Slightly likely

[ ] Moderately likely

[ ] Very likely

[ ] Extremely likely

37. List the three flight symbols that interfere most with the IR imagery.

____

____

____

38. List the three weapons symbols that interfere most with the IR imagery.

____

____

____

39. In the front seat, what percentage of your time during traveling flight do you spend monitoring the PNVS?

__________ percent monitoring the PNVS

40. In the back seat, what percentage of your time during traveling flight do you spend monitoring the TADS?

__________ percent monitoring the TADS

41. In the back seat, what percentage of your time during target engagements do you spend monitoring the TADS?

__________ percent monitoring the TADS

42. At the end of the AQC, how proficient were you in using the PNVS to fly the AH-64?

[ ] Minimally proficient
[ ] Marginally proficient
[ ] Moderately proficient
[ ] Highly proficient
[ ] Extremely proficient

43. At the end of the AQC, how proficient were you in operating the TADS?

[ ] Minimally proficient
[ ] Marginally proficient
[ ] Moderately proficient
[ ] Highly proficient
[ ] Extremely proficient

44. Currently, how proficient are you in flying with the PNVS?

[ ] Minimally proficient
[ ] Marginally proficient
[ ] Moderately proficient
[ ] Highly proficient
[ ] Extremely proficient

45. Currently, how proficient are you in operating the TADS?

[ ] Minimally proficient
[ ] Marginally proficient
[ ] Moderately proficient
[ ] Highly proficient
[ ] Extremely proficient

6th CAVALRY BRIGADE--AIR COMBAT (CBAC)

PART C INSTRUCTIONS: Part C consists of questions that provide information about your personal opinions and preferences. Answer each item that applies to you by checking in the appropriate bracket [√] or by printing your answer in the space provided. This information will be treated as highly confidential; individual responses will not be seen by anyone except the research staff.

46. Which crew station was most difficult for you to learn during the AQC?

    [ ] Front seat
    [ ] Back seat
    [ ] Both seats were equally difficult

47. At the end of the AQC, in which seat did you prefer to be designated if you had to be assigned to only one seat?

    [ ] Front seat
    [ ] Back seat
    [ ] Both seats preferred equally

48. Currently, in which seat would you prefer to be designated if you had to be assigned to only one seat?

    [ ] Front seat
    [ ] Back seat
    [ ] Either seat would be preferred equally

49. Rank order the factors that you believe were considered in making your seat designation. (Put a "1" beside the most important, a "2" beside the next most important, etc. until all factors have been ranked. Put a "0" beside any factors that were not considered. Other than "0," do not use the same number twice.)

    ___________ Needs of the unit (front/back seat manning requirements)
    ___________ Unit policy (e.g., assign all new personnel to front seat)
    ___________ Personal capabilities in the AH-64 as formally evaluated by the unit
    ___________ Personal capabilities in the CMS as formally evaluated by the unit
    ___________ Personal capabilities as evaluated during the AQC
    ___________ Personal preferences
    ___________ Recommendations of unit aviators who knew my capabilities

50. How proficiently could you perform if you were required to occupy your nondesignated crew station in an emergency? (If you are current in both seats, indicate which seat you occupy least often: ____________________; then rate your proficiency in that seat.)

[ ] Not proficient--mission could not be accomplished
[ ] Minimally proficient
[ ] Marginally proficient
[ ] Moderately proficient
[ ] Highly proficient
[ ] Extremely proficient

51. In your opinion, how many hours of refresher training would be required for you to attain RL2 and RL1 status in your non-designated seat?

__________ flight hours to RL2 from RL3
__________ CMS hours to RL2 from RL3
__________ flight hours to RL1 from RL2
__________ CMS hours to RL1 from RL2

52. How adequate is your semiannual familiarization training in the opposite seat?

[ ] Highly inadequate
[ ] Moderately inadequate
[ ] Marginally inadequate
[ ] Marginally adequate
[ ] Moderately adequate
[ ] Highly adequate
[ ] More than adequate

53. To operate effectively as an AH-64 crew, how important is it that you train regularly with the same crewmember?

[ ] Not at all important--the crew only need to be proficient in their own seat
[ ] Slightly important--it is helpful to know how the other crewmember will perform
[ ] Moderately important--regular crew training facilitates crew coordination
[ ] Highly important--regular crew training may affect mission success
[ ] Extremely important--regular crew training is critical to mission success

54. How often do you verbally crosscheck your crewmate to ensure he has completed a prescribed task before you proceed with your tasks?

[ ] Almost never
[ ] Infrequently
[ ] Occasionally
[ ] Frequently
[ ] Almost always

55. Use the following scale to rate the amount of crew communication that is required to perform the mission segments that are listed below, if the crewmembers have never flown together before.

1	2	3	4	5	6	7	8	9
No Crew Communication		Little Crew Communication		Moderate Crew Communication		High Crew Communication		Constant Crew Communication

a. ________ Preflight planning and checks

b. ________ Takeoff and departure

c. ________ Enroute in contour flight (day)

d. ________ Enroute in NOE flight (day)

e. ________ Enroute in contour flight (night using PNVS)

f. ________ Enroute in NOE flight (night using PNVS)

g. ________ Target acquisition (day)

h. ________ Target acquisition (night)

i. ________ Target engagement (day using HELLFIRE)

j. ________ Target engagement (night using HELLFIRE)

k ________ Target engagement (day using rockets)

l. ________ Target engagement (night using rockets)

m. ________ Target engagement (day using 30 mm)

n. ________ Target engagement (night using 30 mm)

56. Use the following scale to rate the amount of crew communication that is required to perform the mission segments that are listed below, if the crewmembers have trained together as a fixed crew for six months.

1	2	3	4	5	6	7	8	9
|	|	|	|	|	|	|	|	|
No Crew Communication		**Little Crew Communication**		**Moderate Crew Communication**		**High Crew Communication**		**Constant Crew Communication**

a. ________ Preflight planning and checks

b. ________ Takeoff and departure

c. ________ Enroute in contour flight (day)

d. ________ Enroute in NOE flight (day)

e. ________ Enroute in contour flight (night using PNVS)

f. ________ Enroute in NOE flight (night using PNVS)

g. ________ Target acquisition (day)

h. ________ Target acquisition (night)

i. ________ Target engagement (day using HELLFIRE)

j. ________ Target engagement (night using HELLFIRE)

k. ________ Target engagement (day using rockets)

l. ________ Target engagement (night using rockets)

m. ________ Target engagement (day using 30 mm)

n. ________ Target engagement (night using 30 mm)

57. Use the following scale to rate the effectiveness of the CMS and CWEPT in training field unit aviators in each seat.

1	2	3	4	5	6	7	8	9
Not Effective		Slightly Effective		Moderately Effective		Highly Effective		Extremely Effective

a. ________ CMS training in the front seat

b. ________ CWEPT training in the front seat

c. ________ CMS training in the back seat

d. ________ CWEPT training in the back seat

58. How many semiannual flight hours do you believe you would need to maintain proficiency in the front seat?

__________ flight hours

__________ CMS hours

59. How many semiannual flight hours do you believe you would need to maintain proficiency in the back seat?

__________ flight hours

__________ CMS hours

60. How many semiannual flight hours do you believe you would need to maintain proficiency in both seats (dual seat currency)?

__________ flight hours

__________ CMS hours

APPENDIX B
POSTFLIGHT DEBRIEFING

AH-64 CMS POST FIELDING TRAINING EFFECTIVENESS ANALYSIS

The following questions refer to the flight that you have just completed and should be answered as soon after the flight as possible. Read each item carefully and answer by checking [√] the appropriate box or by writing in the space provided. Respond to all questions. Regardless of the crew station you occupied, you are to complete one of these forms each time you fly in an AH-64A aircraft, the Combat Mission Simulator (CMS), or the Cockpit Weapons and Emergency Procedural Trainer (CWEPT).

1. What was the **date** of this flight? ____________________

2. What is your full **name and rank**?

____

3. What is the **other crewmember's** full name and rank?

____

4. To which **unit** are you assigned? Circle the appropriate designation.

Squadron:	1/6	3/6	4/6	
Troop:	HHT	A	B	C

5. Which **crew station** did you occupy during this flight? 1[ ] Pilot 2[ ] CPG

6. Were you the **PC** for this flight? 1[ ] Yes 2[ ] No

7. What was the primary **mission** of this flight? [check one]

   1[ ] Satisfy requirements of individual aircrew training program
   2[ ] Satisfy requirements of crew training program
   3[ ] Battle drill
   4[ ] Border mission
   5[ ] Checkride (specify type)__________________________
   6[ ] Other (specify)______________________________

8. Did **more than one aircraft** fly on this mission? 1[ ] Yes 2[ ] No

   If yes,

   a. How many OH-58s? ________

   b. How many AH-64s? ________

9. During this flight, how much **flight time** did you log? _____ hours

10. During this flight, how much **flight time** did you log under the following **flight conditions**?

a. Day ________ hours   d. Terrain ________ hours
b. Hood ________ hours   e. System ________ hours
c. Night ________ hours   f. Weather ________ hours

11. During this flight, how much **flight time** did you log under the following **flight modes**:

a. Contact ________ hours   f. Low-Level ________ hours
b. Tactics ________ hours   g. Contour ________ hours
c. Gunnery ________ hours   h. Formation ________ hours
d. NOE ________ hours   i. Admin. ________ hours
e. Other (specify)________________________ ________ hours

12. Did you receive **target handovers** from another aircraft? 1[ ] Yes 2[ ] No

If yes, how many? ___________ target handovers

13. Was this flight in the AH-64, CMS, or CWEPT?

1[ ] AH-64
2[ ] CMS
3[ ] CWEPT

14. Enter below the number of rounds fired during the flight.

WEAPON SYSTEM	ROUNDS		
	Live	Dry-Fire	Simulated
30mm			
Rockets			
HELLFIRE			

15. In the following table, document the number of times that you practiced specific gunnery tasks on this flight. In the row for each gunnery task that you practiced, enter the **number of times** you employed each (a) sight system, (b) method of range determination, (c) aircraft mode, and (d) target mode. **Include the tasks practiced by both crewmembers on this flight**, not just yourself. If neither crewmember practiced a specific task, enter zero across the row so that each block contains a response. This table must be completed **every time you fly** in the AH-64 or CMS.

WEAPON SYSTEM	SIGHT SYSTEM					RANGE METHOD			AIRCRAFT MODE		TARGET MODE	
	IHADSS		TADS		COOP	LRF	Manual	Hand Over	Hover	Running	Stationary	Moving
	P	G	P	G								
30MM												
Rockets												
HELLFIRE												

APPENDIX C

INSTRUCTOR/OPERATOR SITUATION AND TARGET HANDOVER SHEETS

INSTRUCTOR/OPERATOR SITUATION

REFERENCE: SPECIAL MAP, TODENDORF 1:50,000, REPRODUCTION 1:100,000

CONDITIONS: DAY/PNVS

DURATION: 1.5 HR

SEQUENCE OF EVENTS:

(a) Brief crew as to the conduct of the exercise. Emphasize that weapon selection will be directed by the Scout (Instructor), which is not the norm.

(b) ARI/Instructor will gather the information on page ___ for data collection by crew.

(c) Hostility interrupt will be on.

(d) Crew will conduct day mission, then conduct the same mission under PNVS.

(e) Target engagements will be moving targets from a hover, stationary targets from a hover, and moving targets with the aircraft running fire.

CONDUCT OF THE OPERATION:

(a) Initialize trainer to IC, set 126, insert TEE 318. (Stop all movement of targets.) Crew conducts boresighting (IHADSS and TADS), inserts doppler, present position, and firing points 1 and 2.

(b) Doppler: PPOS (Holding Area) VK86507202; Firing Position 1 - VK84537290; Firing Position 2 - VK84217268.

(c) Crew calls Scout ready.

(d) Move from the holding area to Firing Position 1, BP 22. Give three target handovers.

(e) Move to Firing Position 2, call set.

(f) Move from BP 22 on heading 210° to grid 8369.

(g) Turn right, fly heading 030° to grid 8576. Continue heading to grid 8576.

(h) Turn left to heading 210°, return to BP 22.

(i) Return to holding area.

SCENARIO: TARGET HANDOVERS

(A) FIRING POSITION 1

TGT 1	Type	=	2 T-80 Tanks, Moving South
	Azimuth	=	020°
	Range	=	7000 - 5000 m
	Method	=	Hellfire (LOAL HI, LOBL), CPG, TADS
TGT 2	Type	=	BMP Stationary
	Azimuth	=	015°
	Range	=	4200 m
	Method	=	2.75, COOP, 4 engagements, 1 pair each
TGT 3	Type	=	ZSU Stationary
	Azimuth	=	010°
	Range	=	3900 m
	Method	=	30 mm, CPG, TADS, 20 rounds

(B) FIRING POSITION 2

TGT 1	Type	=	T-80 Tanks, Stationary
	Azimuth	=	280°
	Range	=	2800 m
	Method	=	Hellfire (LOBL), CPG, TADS
TGT 2	Type	=	BMP, Stationary
	Azimuth	=	275°
	Range	=	2200 m
	Method	=	30 mm, PLT, IHADSS, 20 rounds (CPG identify and give handover to pilot)
TGT 3	Type	=	BMP, Stationary
	Azimuth	=	220°
	Range	=	5800 m
	Method	=	2.75, COOP, 4 engagements, 1 pair each

(C) TRAVERSING LOWLAND ROUTE

TGT 4	Type	=	BMP, Moving North
	Azimuth	=	315°
	Range	=	2500 - 1500
	Method	=	30 mm, CPG, TADS, 20 rounds

TGT 5	Type	=	BTR-60, Moving North
	Azimuth	=	240°
	Range	=	3000 - 2000 m
	Method	=	30 mm, CPG, TADS, 20 rounds

(D) FIRING POSITION 1

TGT 6	Type	=	T-80 Tank, Stationary
	Azimuth	=	220°
	Range	=	5000 m
	Method	=	Hellfire (LOAL), CPG, TADS